Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks

Tuyen X. Tran, *Member, IEEE*, and Dario Pompili, *Senior Member, IEEE*

***Abstract*—Mobile-edge computing (MEC) is an emerging paradigm that provides a capillary distribution of cloud computing capabilities to the edge of the wireless access network, enabling rich services and applications in close proximity to the end users. In this paper, an MEC enabled multi-cell wireless network is considered where each base station (BS) is equipped with a MEC server that assists mobile users in executing computation-intensive tasks via task offloading. The problem of joint task offloading and resource allocation is studied in order to maximize the users' task offloading gains, which is measured by a weighted sum of reductions in task completion time and energy consumption. The considered problem is formulated as a mixed integer nonlinear program (MINLP) that involves jointly optimizing the task offloading decision, uplink transmission power of mobile users, and computing resource allocation at the MEC servers. Due to the combinatorial nature of this problem, solving for optimal solution is difficult and impractical for a large-scale network. To overcome this drawback, we propose to decompose the original problem into a resource allocation (RA) problem with fixed task offloading decision and a task offloading (TO) problem that optimizes the optimal-value function corresponding to the RA problem. We address the RA problem using convex and quasi-convex optimization techniques, and propose a novel heuristic algorithm to the TO problem that achieves a suboptimal solution in polynomial time. Simulation results show that our algorithm performs closely to the optimal solution and that it significantly improves the users' offloading utility over traditional approaches.**

***Index Terms*—Mobile edge computing, computation offloading, multi-server resource allocation, distributed systems.**

I. Introduction

THE rapid growth of mobile applications and the Internet of Things (IoTs) have placed severe demands on cloud infrastructure and wireless access networks such as ultra-low latency, user experience continuity, and high reliability. These stringent requirements are driving the need for highly localized services at the network edge in close proximity to the end users. In light of this, the Mobile-Edge Computing (MEC) [1] concept has emerged, which aims at uniting telco, IT, and cloud computing to deliver cloud services directly from the network edge. Differently from traditional cloud computing systems where remote public clouds are utilized, MEC servers are owned by the network operator and are implemented directly at the cellular Base Stations (BSs) or at the local wireless Access Points (APs) using a generic-computing platform. With this position, MEC allows for the execution of applications in close proximity to end users, substantially reducing end-to-end (e2e) delay and releasing the burden on backhaul networks [2].

With the emergence of MEC, the ability of resource-constrained mobile devices to offload computation tasks to the MEC servers is expected to support a myriad of new services and applications such as augmented reality, IoTs, autonomous vehicles and image processing [3]. Example applications such as face detection and recognition for airport security and surveillance [4], video crowd-sourcing that reconstructs multi-view live videos of an event [5], or real-time healthcare data analytics [6] can highly benefit from the collaboration between mobile devices and MEC platform. In the former case, a central authority such as the Federal Bureau of Investigation (FBI) would extend their Amber alerts such that all available cell phones in the area where a missing child was last seen that opt-in to the alert would actively capture images. In the latter case, the MEC servers collect individual input videos (views) captured for the same event from multiple attendees and combine them into multi-view videos, allowing viewers to watch the event from various angles. In both applications, the user devices only provide input data (images, videos) and the computation-intensive tasks (face recognition, multi-view video construction) are then performed at the MEC servers.

Task offloading, however, incurs extra overheads in terms of delay and energy consumption due to the communication required between the devices and the MEC server in the uplink wireless channels. Additionally, in a system with a large number of offloading users, the finite computing resources at the MEC servers considerably affect the task execution delay [7]. Therefore, offloading decisions and performing resource allocation become a critical problem toward enabling efficient computation offloading. Previously, this problem has been partially addressed by optimizing either the offloading decision [7], [8], communication resources [9], [10], or computing resources [11], [12]. Recently, Sardellitti *et al.* [13] addressed the joint allocation of radio and computing resources, while the authors in [14] considered the joint task offloading and resources optimization

Manuscript received September 16, 2017; revised March 7, 2018, May 14, 2018, and September 22, 2018; accepted October 27, 2018. Date of publication November 13, 2018; date of current version January 15, 2019. This work was supported in part by the National Science Foundation under Grant CNS-1319945. The work of Tuyen X. Tran was done while he was a Ph.D. student at Rutgers University. The review of this paper was coordinated by Prof. Y. Cheng. *(Corresponding author: Tuyen Xuan Tran.)*

T. X. Tran is with the AT&T Labs–Research, Bedminster, NJ, USA (e-mail: tuyen.tran@rutgers.edu).

D. Pompili is with the Department of Electrical and Computer Engineering, Rutgers University–New Brunswick, NJ 08901-8554 USA (e-mail: pompili@rutgers.edu).

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

Digital Object Identifier 10.1109/TVT.2018.2881191

0018-9545 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

in a multi-user system. Both of these works, however, only concentrate on a system with a single MEC server.

Our Vision: Unlike the traditional approaches mentioned above, our objective is to design a holistic solution for joint task offloading and resource allocation in a multi-server MEC-assisted network so as to maximize the users' offloading gains. Specifically, we consider a multi-cell ultra-dense network where each BS is equipped with a MEC server to provide computation offloading services to the mobile users. The distributed deployment of the MEC servers along with the densification of (small cell) BSs—as foreseen in the 5G standardization roadmap [15]—will pave the way for real proximity, ultra-low latency access to cloud functionalities. Additionally, the benefits brought by a multi-server MEC system over the single-server MEC (aka single-cloud) system are multi-fold: (i) firstly, as each MEC server may be overloaded when serving a large number of offloading users, one can release the burdens on that server by directing some users to offload to the neighboring servers from the nearby BSs, thus preventing the limited resources on each MEC server from becoming the bottle neck; (ii) secondly, each user can choose to offload its task to the BS with more favorable uplink channel condition, thus saving transmission energy consumption; (iii) finally, coordination of resource allocation to offload users across multiple neighboring BSs can help mitigate the effect of interference and resource contention among the users and hence, improve offloading gains when multiple users offload their tasks simultaneously.

The envisioned scheme requires global knowledge about the computation tasks at the users, the computational resources at the MEC servers, and the wireless channel condition between users and BSs. The mechanism to gather such information can be done similarly as in the Coordinated Multi-Point (CoMP) transmission systems [16] where the BSs exchange their channel information with each other using the logical X2 interface. Alternatively, in the emerging Cloud Radio Access Network (C-RAN) [17], where all the BSs are connected to the same Base Band Unit (BBU) via high-capacity backhaul links, the global system state can be easily accessible at the BBU. Hence, the proposed scheme can be implemented at a central entity, which might be located at an aggregation point in the traditional RAN hierarchy [1] or at the BBU pool in C-RAN. Similarly to the system that performs centralized joint transmission for interference management [16] and/or joint user scheduling [18], our proposed scheme comes with the cost of increased demand on backhaul due to additional signaling overhead. However, it is anticipated that this issue can be overcome when different BSs are connected together like in a C-RAN architecture.

Challenges and Contributions: To exploit in full the benefits of computation offloading in the considered multi-cell, multi-server MEC network, there are several key challenges that need to be addressed. Firstly, the radio resource allocation is much more challenging than the special cases studied in the literature (cf. [14]) due to the presence of inter-cell interference that introduces the coupling among the achievable data rate of different users, which makes the problem nonconvex. Secondly, the complexity of the task-offloading decision is high as, for each user, one needs to decide not only whether it should offload the computation task but also which BS/server to offload the task to. Thirdly, the optimization model should take into account the inherent heterogeneity in terms of mobile devices' computing capabilities, computation task requirements, and availability of computing resources at different MEC servers. In this context, the main contributions of this article are summarized as follows.

1) We model the offloading utility of each user as the weighted-sum of the improvement in task-completion time and device energy consumption; we formulate the problem of Joint Task Offloading and Resource Allocation (JTORA) as a Mixed Integer Non-linear Program (MINLP) that jointly optimizes the task offloading decisions, users' uplink transmit power, and computing resource allocation to offloaded users at the MEC servers, so as to maximize the system offloading utility.
2) While the JTORA problem is very challenging, we propose to decompose the problem into (i) a Resource Allocation (RA) problem with fixed task offloading decision and (ii) a Task Offloading (TO) problem that optimizes the optimal-value function corresponding to the RA problem.
3) We further show that the RA problem can be decoupled into two independent problems, namely the Uplink Power Allocation (UPA) problem and the Computing Resource Allocation (CRA) problem; the resulting UPA and CRA problems are addressed using quasi-convex and convex optimization techniques, respectively.
4) We propose a novel low-complexity heuristic algorithm to tackle the TO problem and show that it achieves a suboptimal solution in polynomial time.
5) We carry out extensive numerical simulations to evaluate the performance of the proposed solution, which is shown to be near-optimal and to improve significantly the users' offloading utility over traditional approaches.

Article Organization: The remainder of this article is organized as follows. In Section II, we review the related works. In Section III, we present the system model. The joint task offloading and resource allocation problem is formulated in Section IV, followed by the decomposition of the problem itself. We present our proposed solution in Section V and numerical results in Section VI. Finally, in Section VII we conclude the article.

II. Related Works

The MEC paradigm has attracted considerable attention in both academia and industry over the past several years. In 2013, Nokia Networks introduced the very first real-world MEC platform [19], in which the computing platform—Radio Applications Cloud Servers (RACS)—is fully integrated with the Flexi Multiradio BS. Saguna also introduced their fully virtualized MEC platform, so called Open-RAN [20], that can provide an open environment for running third-party MEC applications. Recently, a MEC Industry Specifications Group (ISG) was formed to standardize and moderate the adoption of MEC within the RAN [1].

A number of solutions have also been proposed to exploit the potential benefits of MEC in the context of the IoTs and 5G. For instance, our previous work in [2] proposed to explore the

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

synergies among the connected entities in the MEC network and presented three representative use-cases to illustrate the benefits of MEC collaboration in 5G networks. In [21], we proposed a collaborative caching and processing framework in a MEC network whereby the MEC servers can perform both caching and transcoding so as to facilitate Adaptive Bit-Rate (ABR) video streaming. Similar approach was also considered in [22] which combined the traditional client-driven dynamic adaptation scheme, DASH, with network-assisted adaptation capabilities. In addition, MEC is also seen as a key enabling technique for connected vehicles by adding computation and geo-distributed services to the roadside BSs so as to analyze the data from proximate vehicles and roadside sensors and to propagate messages to the drivers in very low latency [23].

Recently, several works have focused on exploiting the benefits of computation offloading in MEC network [24]. Note that similar problems have been investigated in conventional Mobile Cloud Computing (MCC) systems [25]. However, a large body of existing works on MCC assumed an infinite amount of computing resources available in the cloudlets, where the offloaded tasks can be executed with negligible delay [26]–[28]. The problem of offloading scheduling was then reduced to radio resource allocation in [9] where the competition for radio resources is modeled as a congestion game of selfish mobile users. In the context of MEC, the problem of joint task offloading and resource allocation was studied in a single-user system with energy harvesting devices [29], and in a multi-cell multi-user systems [13]; however the congestion of computing resources at the MEC server was omitted. Similar problem is studied in [14] considering the limited edge computing resources in a single-server MEC system.

Computation offloading in MEC can also utilize the heterogeneous resource pool constituted by the end-user devices. In our previous work [2], a novel resource management framework was envisioned to orchestrate both the *horizontal* collaboration at the end-user layer and at the MEC layer as well as the *vertical* collaboration among end-users, edge nodes, and cloud nodes. Along this line, Chen *et al.* [30] proposed a Device-to-Device (D2D) Crowd framework where a massive crowd of devices at the network edge leverage network-assisted D2D collaboration for computation and communication resource sharing. Using a game-theoretic approach, the work in [31] considered a computation offloading problem that involves selfish mobile users that want to maximize their individual performance. However, the difference between [30], [31] and our work is multi-fold. Firstly, the focus of [30], [30] is on the task assignment problem in which the data rates of the wireless links are considered as constants, whereas our work considers the joint design of task assignment and resource allocation policy for which the varying wireless channels and multi-user interference are taken into account when calculating the data rates. Secondly, the objective in [30] is to minimize the total energy consumption on mobile devices while we consider a parameterized objective function that can be tuned to optimize both the energy consumption and task execution time. Thirdly, it is assumed in [31] that the users always use a constant transmit power while our approach optimizes users' transmit power.

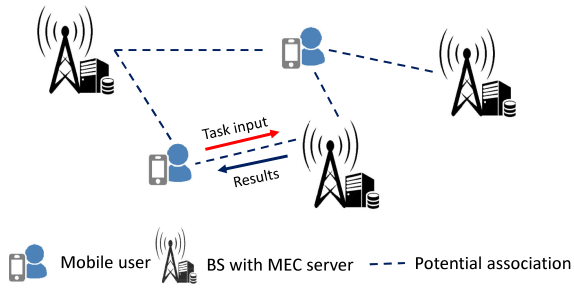


Fig. 1. Example of a cellular system with MEC servers deployed at the BSs.

In summary, most of the existing works did not consider a holistic approach that jointly determines the task offloading decision and the radio and computing resource allocation in a multi-cell, multi-server system as considered in this article.

III. System Model

We consider a multi-cell, multi-server MEC system as illustrated in Fig. 1, in which each BS is equipped with a MEC server to provide computation offloading services to the resource-constrained mobile users such as smart phones, tablets, and wearable devices. In general, each MEC server can be either a physical server or a virtual machine with moderate computing capabilities provisioned by the network operator and can communicate with the mobile devices through wireless channels provided by the corresponding BS. Each mobile user can choose to offload computation tasks to a MEC server from one of the nearby BSs it can connect to. We denote the set of users and MEC servers in the mobile system as $\mathcal{U} = \{1, 2, \ldots, U\}$ and $\mathcal{S} = \{1, 2, \ldots, S\}$, respectively. For ease of presentation, we will refer to the MEC server s and BS s interchangeably. The modeling of user computation tasks, task uploading transmissions, MEC computation resources, and offloading utility are presented here below. For ease of reference, the key notations used in the article are summarized in Table I.

A. User Computation Tasks

We consider that each user $u \in \mathcal{U}$ has one computation task at a time, denoted as T_u, that is atomic and cannot be divided into subtasks. Each computation task T_u is characterized by a tuple of two parameters, $\langle d_u, c_u \rangle$, in which d_u [bits] specifies the amount of input data necessary to transfer the program execution (including system settings, program codes, and input parameters) from the local device to the MEC server, and c_u [cycles] specifies the workload, i.e., the amount of computation to accomplish the task. The values of d_u and c_u can be obtained through carefully profiling of the task execution [7], [32]. Each task can be performed locally on the user device or offloaded to a MEC server. By offloading the computation task to the MEC server, the mobile user would save its energy for task execution; however, it would consume additional time and energy for sending the task input in the uplink.

Let $f_u^l > 0$ denote the local computing capability of user u in terms of CPU cycles/s. Hence, if user u executes its task locally, the task completion time is $t_u^l = \frac{c_u}{f_u^l}$ [seconds]. To

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	Description
$\mathcal{U}$	Set of U users
$\mathcal{S}$	Set of S BSs/MEC servers
T_u	Computation task of user u
d_u	Input data of computation task T_u
c_u	Workload of computation task T_u
f_u^l	Local computing capability of user u
E_u^l	Energy consumption of user u when executing its task locally
E_u	Energy consumption of user u when offloading its task
κ	Energy coefficient of user device
t_u^l	Local execution time of task T_u
t_{up}^u	Transmission time of task T_u to the MEC server
t_{exe}^u	Execution time of task T_u at the MEC server
t_u	Delay experienced by user u when offloading its task
B	Uplink system bandwidth
$\mathcal{N}$	Set of N orthogonal sub-bands
x_{us}^j	Task offloading indicator, $\forall u \in \mathcal{U}, s \in \mathcal{S}, j \in \mathcal{N}$
$\mathcal{G}$	Task offloading ground set
$\mathcal{X}$	Task offloading policy
$\mathcal{U}_{\text{Off}}$	Set of all users that offload their tasks
$\mathcal{U}_s$	Set of users that offload their tasks to server s
h_{us}^j	Uplink channel gain between user u and BS s on sub-band j
p_u	Transmission power of user u
P_u	Maximum transmission power of user u
γ_{us}^j	SINR from user u to BS s on sub-band j
R_{us}	Uplink data rate from user u to BS s
f_s	Computing capacity of server s
f_{su}	Computing resource that server s allocates to task of user u
$\mathcal{F}$	Computing resource allocation policy
J_u	Offloading utility of user u
β_u^t	Preference of user u on task completion time
β_u^e	Preference of user u on energy consumption
λ_u	Resource provider's preference towards user u

calculate the energy consumption of a user device when executing its task locally, we use the widely adopted model of the energy consumption per computing cycle as $\mathcal{E} = \kappa f^2$ [9], [33], where κ is the energy coefficient depending on the chip architecture and f is the CPU frequency. Thus, the energy consumption, E_u^l $[J]$, of user u when executing its task T_u locally, is calculated as,

$$E_u^l = \kappa \left(f_u^l\right)^2 c_u. \tag{1}$$

B. Task Uploading

In case user u offloads its task T_u to one of the MEC servers, the incurred delay comprises: (i) the time t_{up}^u [s] to transmit the input to the MEC server on the uplink, (ii) the time t_{exe}^u [s] to execute the task at the MEC server, and (iii) the time to transmit the output from the MEC server back to the user on the downlink. Since the size of the output is generally much smaller than the input, plus the downlink data rate is much higher than that of the uplink, we omit the delay of transferring the output in our computation, as also considered in [9], [14], [31]. Note that, when the delay of the downlink transmission of output data is non-negligible, our proposed algorithm can still be directly applied for a given downlink rate allocation scheme and known output data size.

In this work, we consider the system with OFDMA as the multiple access scheme in the uplink [34], in which the operational frequency band B is divided into N equal sub-bands of size $W = B/N$ [Hz]. To ensure the orthogonality of uplink transmissions among users associated with the same BS, each user is assigned to one sub-band. Thus, each BS can serve at most N users at the same time. Let $\mathcal{N} = \{1, \ldots, N\}$ be the set of available sub-band at each BS. We define the task offloading variables, which also incorporate the uplink sub-band scheduling, as $x_{us}^j, u \in \mathcal{U}, s \in \mathcal{S}, j \in \mathcal{N}$, where $x_{us}^j = 1$ indicates that task T_u from user u is offloaded to BS s on sub-band j, and $x_{us}^j = 0$ otherwise. We define the ground set $\mathcal{G}$ that contains all the task offloading variables as $\mathcal{G} = \left\{x_{us}^j \,|u \in \mathcal{U}, s \in \mathcal{S}, j \in \mathcal{N}\right\}$ and the task offloading policy $\mathcal{X}$ expressed as $\mathcal{X} = \left\{x_{us}^j \in \mathcal{G} \,\middle|\, x_{us}^j = 1\right\}$. As each task can be either executed locally or offloaded to at most one MEC server, a feasible offloading policy must satisfy the constraint below,

$$\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{N}} x_{us}^j \leq 1, \ \forall u \in \mathcal{U}. \tag{2}$$

Additionally, we denote $\mathcal{U}_s = \{u \in \mathcal{U} | \sum_{j \in \mathcal{N}} x_{us}^j = 1\}$ as the set of users offloading their tasks to server s, and $\mathcal{U}_{\text{off}} = \bigcup_{s \in \mathcal{S}} \mathcal{U}_s$ as the set of users that offload their tasks.

Furthermore, we consider that each user and BS have a single antenna for uplink transmissions (as also considered in [14], [35]). Extension to the case where each BS uses multiple antennas for receiving uplink signals will be addressed in a future work. Denote h_{us}^j as the uplink channel gain between user u and BS s on sub-band j, which captures the effect of path-loss, shadowing, and antenna gain. Note that the user-BS association usually takes place in a large time scale (duration of an offloading session) that is much larger than the time scale of small-scale fading. Hence, similar to [18], we consider that the effect of fast-fading is averaged out during the association. Let $\mathcal{P} = \{p_u \,|0 < p_u \leq P_u, u \in \mathcal{U}_{\text{off}}\}$ denote the users' transmission power, where p_u [W] is the transmission power of user u when uploading its task's input d_u to the BS, subject to a maximum budget P_u. Note that $p_u = 0, \forall u \notin \mathcal{U}_{\text{off}}$. As the users transmitting to the same BS use different sub-bands, the uplink intra-cell interference is well mitigated; still, these users suffer from the inter-cell interference. In this case, the Signal-to-Interference-plus-Noise Ratio (SINR) from user u to BS s on sub-band j is given by,

$$\gamma_{us}^j = \frac{p_u h_{us}^j}{\sum_{r \in \mathcal{S} \setminus \{s\}} \sum_{k \in \mathcal{U}_r} x_{kr}^j p_k h_{ks}^j + \sigma^2}, \quad \forall u \in \mathcal{U}, s \in \mathcal{S}, j \in \mathcal{N}, \tag{3}$$

where σ^2 is the background noise variance and the first term at the denominator is the accumulated intra-cell interference from all the users associated with other BSs on the same sub-band j. Since each user only transmits on one sub-band, the achievable rate [bits/s] of user u when sending data to BS s is given as,

$$R_{us}\left(\mathcal{X}, \mathcal{P}\right) = W \log_2\left(1 + \gamma_{us}\right), \tag{4}$$

where $\gamma_{us} = \sum_{j \in \mathcal{N}} \gamma_{us}^j$. Moreover, let $x_{us} = \sum_{j \in \mathcal{N}} x_{us}^j, \forall u \in \mathcal{U}, s \in \mathcal{S}$. Hence, the transmission time of user u when sending its task input d_u in the uplink can be calculated as,

$$t_{\text{up}}^u = \sum_{s \in \mathcal{S}} \frac{x_{us} d_u}{R_{us}\left(\mathcal{X}, \mathcal{P}\right)}, \forall u \in \mathcal{U}. \tag{5}$$

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

C. MEC Computing Resources

The MEC server at each BS is able to provide computation offloading service to multiple users concurrently. The computing resources made available by each MEC server to be shared among the associating users are quantified by the computational rate f_s, expressed in terms of number of CPU cycles/s. After receiving the offloaded task from a user, the server will execute the task on behalf of the user and, upon completion, will return the output result back to the user. We define the computing resource allocation policy as $\mathcal{F} = \{f_{us} \,|u \in \mathcal{U}, s \in \mathcal{S}\}$, in which f_{us} [cycles/s] > 0 is the amount of computing resource that BS s allocates to task T_u offloaded from user $u \in \mathcal{U}_s$. Hence, clearly $f_{us} = 0, \forall u \notin \mathcal{U}_s$. In addition, a feasible computing resource allocation policy must satisfy the computing resource constraint, expressed as,

$$\sum_{u \in \mathcal{U}} f_{us} \leq f_s, \forall s \in \mathcal{S}. \tag{6}$$

Given the computing resource assignment $\{f_{us}, s \in \mathcal{S}\}$, the execution time of task T_u at the MEC servers is,

$$t^u_{\text{exe}} = \sum_{s \in \mathcal{S}} \frac{x_{us} c_u}{f_{us}}, \forall u \in \mathcal{U}. \tag{7}$$

D. User Offloading Utility

Given the offloading policy $\mathcal{X}$, the transmission power p_u, and the computing resource allocation f_{us}'s, the total delay experienced by user u when offloading its task is given by,

$$t_u = t^u_{\text{up}} + t^u_{\text{exe}} = \sum_{s \in \mathcal{S}} x_{us} \left(\frac{d_u}{R_{us}(\mathcal{X}, \mathcal{P})} + \frac{c_u}{f_{us}} \right), \forall u \in \mathcal{U}. \tag{8}$$

The energy consumption of user u, E_u [J], due to uploading transmission is calculated as $E_u = \frac{p_u t^u_{\text{up}}}{\xi_u}, \forall u \in \mathcal{U}$, where ξ_u is the power amplifier efficiency of user u. Without loss of generality, we assume that $\xi_u = 1, \forall u \in \mathcal{U}$. Thus, the uplink energy consumption of user u simplifies to,

$$E_u = p_u t^u_{\text{up}} = p_u d_u \sum_{s \in \mathcal{S}} \frac{x_{us}}{R_{us}(\mathcal{X}, \mathcal{P})}, \forall u \in \mathcal{U}. \tag{9}$$

In a mobile cloud computing system, the users' QoE is mainly characterized by their task completion time and energy consumption. In the considered scenario, the relative improvement in task completion time and energy consumption are characterized by $\frac{t^l_u - t_u}{t^l_u}$ and $\frac{E^l_u - E_u}{E^l_u}$, respectively [14]. Therefore, we define the offloading utility of user u as,

$$J_u = \left(\beta^t_u \frac{t^l_u - t_u}{t^l_u} + \beta^e_u \frac{E^l_u - E_u}{E^l_u} \right) \sum_{s \in \mathcal{S}} x_{us}, \forall u \in \mathcal{U}, \tag{10}$$

in which $\beta^t_u, \beta^e_u \in [0, 1]$, with $\beta^t_u + \beta^e_u = 1, \forall u \in \mathcal{U}$, specify user u's preference on task completion time and energy consumption, respectively. For example, a user u with short battery life can increase β^e_u and decrease β^t_u so as to save more energy at the expense of longer task completion time. In practice, the value of β^e_u can be set by the mobile user through different power saving modes; for instance, $\beta^e_u = 1$ at "extreme power saving" mode and $\beta^e_u = 0$ at "maximum performance" mode. Alternatively, β^t_u can be set proportionally to the battery percentage level of the device. Note that offloading too many tasks to the MEC servers will cause excessive delay due to the limited bandwidth and computing resources at the MEC servers, and consequently degrade some users' QoE compared to executing their tasks locally. Hence, clearly user u should not offload its task to the MEC servers if $J_u \leq 0$.

The expressions of the task completion time and energy consumption in (10) clearly show the interplay between radio access and computational aspects, which motivates a joint optimization of offloading scheduling, radio, and computing resources so as to optimize users' offloading utility.

IV. Problem Formulation

We formulate here the problem of joint task offloading and resource allocation, followed by the outline of our decomposition approach.

A. Joint Task Offloading and Resource Allocation Problem

For a given offloading decision $\mathcal{X}$, uplink power allocation $\mathcal{P}$, and computing resource allocation $\mathcal{F}$, we define the system utility as the weighted-sum of all the users' offloading utilities,

$$J(\mathcal{X}, \mathcal{P}, \mathcal{F}) = \sum_{u \in \mathcal{U}} \lambda_u J_u, \tag{11}$$

with J_u given in (10) and $\lambda_u \in (0, 1]$ specifying the resource provider's preference towards user u, $\forall u \in \mathcal{U}$. For instance, depending on the payments offered by the users, the resource provider could prioritize users with higher revenues for offloading by increasing their corresponding preferences. Additionally, λ_u can also be set based on user type and criticality of the computation task. For example, computation tasks from mobile devices involved in public safety operations (e.g., devices carried by police officers and first responders) should be prioritized with high value of λ_u. We now formulate the Joint Task Offloading and Resource Allocation (JTORA) problem as a system utility maximization problem, i.e.,

$$\max_{\mathcal{X}, \mathcal{P}, \mathcal{F}} \quad J(\mathcal{X}, \mathcal{P}, \mathcal{F}) \tag{12a}$$

$$\text{s.t.} \quad x^j_{us} \in \{0, 1\}, \forall u \in \mathcal{U}, s \in \mathcal{S}, j \in \mathcal{N}, \tag{12b}$$

$$\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{N}} x^j_{us} \leq 1, \forall u \in \mathcal{U}, \tag{12c}$$

$$\sum_{u \in \mathcal{U}} x^j_{us} \leq 1, \forall s \in \mathcal{S}, j \in \mathcal{N}, \tag{12d}$$

$$0 < p_u \leq P_u, \forall u \in \mathcal{U}_{\text{off}}, \tag{12e}$$

$$f_{us} > 0, \forall u \in \mathcal{U}_s, s \in \mathcal{S}, \tag{12f}$$

$$\sum\nolimits_{u \in \mathcal{U}} f_{us} \leq f_s, \forall s \in \mathcal{S}. \tag{12g}$$

The constraints in the formulation above can be explained as follows: constraints (12b) and (12c) imply that each task can be either executed locally or offloaded to at most one server on

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

one sub-band; constraint (12d) implies that each BS can serve at most one user per sub-band; constraint (12e) specifies the transmission power budget of each user; finally, constraints (12f) and (12g) state that each MEC server must allocate a positive computing resource to each user associated with it and that the total computing resources allocated to all the associated users must not excess the server's computing capacity. The JTORA problem in (12) is a Mixed Integer Nonlinear Program (MINLP) and finding the optimal solution usually requires exponential time complexity [36]. Given the large number of variables that scale linearly with the number of users, MEC servers, and sub-bands, our goal is to design a low-complexity, suboptimal solution that achieves competitive performance while being practical to implement.

B. Problem Decomposition

By exploiting the structure of the objective function and constraints in the formulation of JTORA problem in (12), we observe that by temporarily fixing the binary variables $\{x_{us}\}$, problem (12) can be decomposed into multiple subproblems with separated objective and constraints. Leveraging this characteristic and motivated by the approach in [37], we can employ the Tammer decomposition method [38] to transform the original problem with high complexity into an equivalent *master problem* and a set of *subproblems* with lower complexity. Firstly, we rewrite the JTORA problem in (12) as,

$$\max_{\mathcal{X}} \left(\max_{\mathcal{P},\mathcal{F}} J\left(\mathcal{X},\mathcal{P},\mathcal{F}\right) \right) \tag{13a}$$

$$\text{s.t.} \quad \text{(12b)–(12g)}. \tag{13b}$$

Note that the constraints on the offloading decision, $\mathcal{X}$, in (12b), (12c), (12d), and the RA policies, $\mathcal{P},\mathcal{F}$, in (12e), (12f), (12g), are decoupled from each other; therefore, solving the problem in (13) is equivalent to solving the following Task Offloading (TO) problem,

$$\max_{\mathcal{X}} J^*\left(\mathcal{X}\right) \tag{14a}$$

$$\text{s.t.} \quad \text{(12b), (12c), (12d)}, \tag{14b}$$

in which $J^*\left(\mathcal{X}\right)$ is the optimal-value function corresponding to the RA problem, written as,

$$J^*\left(\mathcal{X}\right) = \max_{\mathcal{P},\mathcal{F}} J\left(\mathcal{X},\mathcal{P},\mathcal{F}\right) \tag{15a}$$

$$\text{s.t.} \quad \text{(12e), (12f), (12g)}, \tag{15b}$$

Note that the decomposition from problem (12) to problems (14) and (15) does not change the optimality of the solution [38]. In the next section, we will present our solutions to both the RA problem and the TO problem so as to finally obtain the solution to the original JTORA problem.

V. Low-Complexity Algorithm for Joint Task Offloading and Resource Allocation

We present now our low-complexity approach to solve the JTORA problem by solving first the RA problem in (15) and then using its solution to derive the solution of the TO problem in (14).

Firstly, given a feasible task offloading decision $\mathcal{X}$ that satisfies constraints (12b), (12c), and (12d), and using the expression of J_u in (10), the objective function in (15a) can be rewritten as,

$$J\left(\mathcal{X},\mathcal{P},\mathcal{F}\right) = \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}_s} \lambda_u \left(\beta_u^t + \beta_u^e\right) - V\left(\mathcal{X},\mathcal{P},\mathcal{F}\right), \tag{16}$$

$$\text{where} \quad V\left(\mathcal{X},\mathcal{P},\mathcal{F}\right) = \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}_s} \lambda_u \left(\frac{\beta_u^t t_u}{t_u^l} + \frac{\beta_u^e E_u}{E_u^l}\right). \tag{17}$$

We observe that the first term on the right hand side (RHS) of (16) is constant for a particular offloading decision, while $V\left(\mathcal{X},\mathcal{P},\mathcal{F}\right)$ can be seen as the total offloading overheads of all offloaded users. Hence, we can recast (15) as the problem of minimizing the total offloading overheads, i.e.,

$$\min_{\mathcal{P},\mathcal{F}} V\left(\mathcal{X},\mathcal{P},\mathcal{F}\right) \tag{18a}$$

$$\text{s.t.} \quad \text{(12e), (12f), (12g)}. \tag{18b}$$

Furthermore, from (8), (9), and (17), we have,

$$V\left(\mathcal{X},\mathcal{P},\mathcal{F}\right) = \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}_s} \frac{\phi_u + \psi_u p_u}{\log_2\left(1+\gamma_{us}\right)} + \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}_s} \frac{\eta_u}{f_{us}}, \tag{19}$$

in which, for simplicity, $\phi_u = \frac{\lambda_u \beta_u^t d_u}{t_u^l W}$, $\psi_u = \frac{\lambda_u \beta_u^e d_u}{E_u^l W}$, and $\eta_u = \lambda_u \beta_u^t f_u^l$. Notice from (18b) and (19) that the problem in (18) has a *separable structure*, i.e., the objectives and constraints corresponding to the power allocation p_u's and computing resource allocation f_{us}'s can be decoupled from each other. Leveraging this property, we can decouple problem (18) into two independent problems, namely the *Uplink Power Allocation (UPA)* and the *Computing Resource Allocation (CRA)*, and address them separately, as described in the following sections.

A. Uplink Power Allocation (UPA)

The UPA problem is decoupled from problem (18) by considering the first term on the RHS of (19) as the objective function. Specifically, the UPA problem is expressed as,

$$\min_{\mathcal{P}} \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}_s} \frac{\phi_u + \psi_u p_u}{\log_2\left(1+\gamma_{us}\right)} \tag{20a}$$

$$\text{s.t.} \quad 0 < p_u \le P_u, \forall u \in \mathcal{U}. \tag{20b}$$

Problem (20) is non-convex and difficult to solve because the uplink SINR γ_{us}^j corresponding to user $u \in \mathcal{U}_s$ depends on the transmit power of the other users associated with other BSs on the same sub-band j through the inter-cell interference $I_{us}^j = \sum_{w\in\mathcal{S}\setminus\{s\}}\sum_{k\in\mathcal{U}_w} x_{ks}^j p_k h_{ks}^j$, as seen in (3). Our approach is to find an approximation for I_{us}^j and thus for γ_{us}^j such that problem (20) can be decomposed into sub-problems that, in turn, can be efficiently solved. The optimal uplink power allocation $\mathcal{P}^*$ still generates small objective value for (20).

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

Suppose each BS $s \in \mathcal{S}$ calculates its uplink power allocation independently, i.e., without mutual cooperation, and informs its associated users about the uplink transmit power; then, an achievable upper bound for I_{us}^{j} is given by,

$$\tilde{I}_{us}^{j} \triangleq \sum_{w \in \mathcal{S} \setminus \{s\}} \sum_{k \in \mathcal{U}_w} x_{ks}^{j} P_k h_{ks}^{j}, \forall u \in \mathcal{U}_s, s \in \mathcal{S}, j \in \mathcal{N}. \quad (21)$$

Similar to [39], we argue that $\tilde{I}_{us}^{j}$ is a good estimate of I_{us}^{j} since our offloading decision $\mathcal{X}$ is geared towards choosing the appropriate user-BS associations so as that $\tilde{I}_{us}^{j}$ be small in the first place. This means that a small error in I_{us}^{j} should not lead to large bias in γ_{us}^{j} [39].

By replacing I_{us}^{j} with $\tilde{I}_{us}^{j}$, we get the approximation for the uplink SINR for user u uploading to BS s on sub-band j as,

$$\tilde{\gamma}_{us}^{j} = \frac{p_u h_{us}^{j}}{\tilde{I}_{us}^{j} + \sigma^2}, \forall u \in \mathcal{U}_s, s \in \mathcal{S}, j \in \mathcal{N}. \quad (22)$$

Let $\vartheta_{us} = \sum_{j \in \mathcal{N}} h_{us}^{j} / (\tilde{I}_{us}^{j} + \sigma^2)$ and $\Gamma_s(p_u) = \frac{\phi_u + \psi_u p_u}{\log_2(1 + \vartheta_{us} p_u)}$. The objective function in (20a) can now be approximated by $\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \Gamma_s(p_u)$. With this position, it can be seen that the objective function and the constraint corresponding to each user's transmit power is now decoupled from each other. Therefore, the UPA problem in (20) can be approximated by $\sum_{s \in \mathcal{S}} |\mathcal{U}_s|$ sub-problems, each optimizing the transmit power of a user $u \in \mathcal{U}_s, s \in \mathcal{S}$, and can be written as,

$$\min \sum_{u \in \mathcal{U}_s} \Gamma_s(p_u) \quad (23a)$$

$$\text{s.t.} \quad 0 < p_u \leq P_u. \quad (23b)$$

Problem (23) is still non-convex as the second-order derivative of the objective function with respect to (w.r.t) p_u, i.e., $\Gamma_s''(p_u)$, is not always positive. However, we can employ quasi-convex optimization technique to address problem (23) based on the following lemma.

Lemma 1: $\Gamma_s(p_u)$ is strictly quasi-convex in the domain defined in (23b).

Proof: See Appendix A. ■

In general, a quasi-convex problem can be solved using the bisection method, which solves a convex feasibility problem in each iteration [40]. However, the popular interior cutting plane method for solving a convex feasibility problem requires $\mathcal{O}(n^2/\epsilon^2)$ iterations, where n is the dimension of the problem. We now propose to further reduce the complexity of the bisection method.

Firstly, notice that a quasi-convex function achieves a local optimum at the diminishing point of the first-order derivative, and that any local optimum of a strictly quasi-convex function is the global optimum [41]. Therefore, based on Lemma 1, we can confirm that the optimal solution p_u^* of problem (23) either lies at the constraint border, i.e., $p_u^* = P_u$ or satisfies $\Gamma_s'(p_u^*) = 0$. It can be verified that $\Gamma_s'(p_u) = 0$ when,

$$\Omega_s(p_u) = \psi_u \log_2(1 + \vartheta_{us} p_u) - \frac{\vartheta_{us}(\phi_u + \psi_u p_u)}{(1 + \vartheta_{us} p_u)\ln 2} = 0. \quad (24)$$

Algorithm 1: Bisection Method for Uplink Power Allocation.

1: Calculate $\Omega_s(P_u)$ using (24)
2: **if** $\Omega_s(P_u) \leq 0$ **then**
3: $\quad p_u^* = P_u$
4: **else**
5: $\quad$ Set optimality tolerance $\epsilon > 0$
6: $\quad$ Initialize $p_u' = 0$ and $p_u'' = P_u$
7: $\quad$ **repeat**
8: $\quad\quad$ Set $p_u^* = (p_u' + p_u'')/2$
9: $\quad\quad$ **if** $\Omega_s(p_u^*) \leq 0$ **then**
10: $\quad\quad\quad$ Set $p_u' = p_u^*$
11: $\quad\quad$ **else**
12: $\quad\quad\quad$ Set $p_u'' = p_u^*$
13: $\quad\quad$ **end if**
14: $\quad$ **until** $p_u'' - p_u' \leq \xi$
15: $\quad$ Set $p_u^* = (p_u' + p_u'')/2$
16: **end if**

Moreover, we have, $\Omega_s'(p_u) = \frac{\vartheta_{us}^2(\phi_u + \psi_u p_u)}{(1 + \vartheta_{us} p_u)^2 \ln 2} > 0$, and $\Omega_s(0) = -\frac{\vartheta_{us}\phi_u}{\ln 2} < 0$. This implies that $\Omega_s(p_u)$ is a monotonically increasing function and is negative at the starting point $p_u = 0$. Therefore, we can design a low-complexity bisection method that evaluates $\Omega_s(p_u)$ in each iteration instead of solving a convex feasibility problem, so as to obtain the optimal solution p_u^*, as presented in Algorithm 1.

In Algorithm 1, if $\Omega_s(P_u) > 0$, the algorithm will terminate in exactly $\lceil \log_2(P_u/\xi) \rceil$ iterations, where ξ is the convergence threshold in line 14. Let $\mathcal{P}^* = \{p_u^*, u \in \mathcal{U}\}$ denote the optimal uplink transmit power policy for a given task offloading policy $\mathcal{X}$. In addition, we denote now as $\Gamma(\mathcal{X}, \mathcal{P}^*)$ the objective value of problem (20) corresponding to $\mathcal{P}^*$.

B. Computing Resource Allocation (CRA)

The CRA problem optimizes the second term on the RHS of (19) and is expressed as follows,

$$\min_{\mathcal{F}} \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \eta_u / f_{us} \quad (25a)$$

$$\text{s.t.} \quad \sum_{u \in \mathcal{U}} f_{us} \leq f_s, \forall s \in \mathcal{S}, \quad (25b)$$

$$f_{us} > 0, \forall u \in \mathcal{U}_s, s \in \mathcal{S}. \quad (25c)$$

Notice that the constraints in (25b) and (25c) are convex. Denote the objective function in (25a) as $\Lambda(\mathcal{X}, \mathcal{F})$; by calculating the second-order derivatives of $\Lambda(\mathcal{X}, \mathcal{F})$ w.r.t. f_{us}, we have,

$$\frac{\partial^2 \Lambda(\mathcal{X}, \mathcal{F})}{\partial f_{us}^2} = \frac{2\eta_u}{f_{us}^3} > 0, \forall s \in \mathcal{S}, u \in \mathcal{U}_s, \quad (26a)$$

$$\frac{\partial^2 \Lambda(\mathcal{X}, \mathcal{F})}{\partial f_{us} \partial f_{vw}} = 0, \forall (u, s) \neq (v, w). \quad (26b)$$

It can be seen that the Hessian matrix of the objective function in (25a) is diagonal with the strictly positive elements, thus it is positive-definite. Hence, (25) is a convex optimization

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

problem and can be solved using Karush-Kuhn-Tucker (KKT) conditions. We have the following Lemma.

Lemma 2: The optimal computing resource allocation f_{us}^* for problem (25) and the corresponding optimal objective function $\Lambda\left(\mathcal{X}, \mathcal{F}^*\right)$ are given, respectively, as,

$$f_{us}^* = \frac{f_s \sqrt{\eta_u}}{\sum_{u \in \mathcal{U}_s} \sqrt{\eta_u}}, \forall s \in \mathcal{S}, u \in \mathcal{U}_s, \tag{27}$$

$$\Lambda\left(\mathcal{X}, \mathcal{F}^*\right) = \sum_{s \in \mathcal{S}} \frac{1}{f_s} \left(\sum_{u \in \mathcal{U}_s} \sqrt{\eta_u} \right)^2. \tag{28}$$

Proof: See Appendix B. ■

C. Joint Task Offloading Scheduling and Resource Allocation

In the previous sections, for a given task offloading decision $\mathcal{X}$, we obtained the solutions for the radio and computing resources allocation. In particular, according to (15), (16), (19), and (28), we have,

$$J^*\left(\mathcal{X}\right) = \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \lambda_u \left(\beta_u^t + \beta_u^e\right) - \Gamma\left(\mathcal{X}, \mathcal{P}^*\right) - \Lambda\left(\mathcal{X}, \mathcal{F}^*\right), \tag{29}$$

where $\mathcal{P}^*$ can be obtained through Algorithm 1 and $\Lambda\left(\mathcal{X}, \mathcal{F}^*\right)$ can be calculated using the closed-form expression in (28). Now, using (29), we can rewrite the TO problem in (14) as,

$$\max_{\mathcal{X}} \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \lambda_u \left(\beta_u^t + \beta_u^e\right) - \Gamma\left(\mathcal{X}, \mathcal{P}^*\right) - \Lambda\left(\mathcal{X}, \mathcal{F}^*\right) \tag{30a}$$

$$\text{s.t.} \quad x_{us}^j \in \{0,1\}, \forall u \in \mathcal{U}, s \in \mathcal{S}, j \in \mathcal{N}, \tag{30b}$$

$$\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{N}} x_{us}^j \leq 1, \forall u \in \mathcal{U}, \tag{30c}$$

$$\sum_{u \in \mathcal{U}} x_{us}^j \leq 1, \forall s \in \mathcal{S}, j \in \mathcal{N}. \tag{30d}$$

Problem (30) consists in maximizing a set function $J^*\left(\mathcal{X}\right)$ w.r.t $\mathcal{X}$ over the ground set $\mathcal{G}$ defined by (30b), and the constraints in (30c) and (30d) define two matroids over $\mathcal{G}$.[1]

Given the combinatorial nature of the TO problem, solving for an optimal solution in polynomial time is extremely challenging. One straightforward approach to solve problem (30) is to use exhaustive search method over all possible task offloading decisions. However, the total number of candidate task offloading decisions would be 2^n where $n = S \times U \times N$. Hence, the exhaustive search method is clearly impractical.

To overcome the aforementioned drawback, we propose a low-complexity heuristic algorithm that can find a local optimum to problem (30) in polynomial time. Specifically, our algorithm starts with an empty set $\mathcal{X} = \emptyset$ and repeatedly performs one of the local operations, namely the *remove* operation or the *exchange* operation, as described in Routine 1, if it improves the set value $J^*(\mathcal{X})$. As we are dealing with two matroid constraints, the *exchange* operation involves adding one element from outside of the current set and dropping up to 2 elements from the set, so as to comply with the constraints. In summary, our proposed heuristic algorithm for task offloading scheduling is presented in Algorithm 2.

[1]For detailed definition of matroid constraint on set function, refer to [42].

Routine 1: Remove and Exchange Operations.

$remove\left(\mathcal{X}, x_{us}^j\right)$
Set $\mathcal{X} \leftarrow \mathcal{X} \backslash \left\{x_{us}^j\right\}$
Output: $\mathcal{X}$
$exchange\left(\mathcal{X}, x_{us}^j\right)$
for $w \in \mathcal{S}, i \in \mathcal{N}$ **do** ▷ to comply with (30c)
  $\mathcal{X} \leftarrow \mathcal{X} \backslash \left\{x_{uw}^i\right\}$
end for
for $v \in \mathcal{U}$ **do** ▷ to comply with (30d)
  $\mathcal{X} \leftarrow \mathcal{X} \backslash \left\{x_{vs}^j\right\}$
end for
Set $\mathcal{X} \leftarrow \mathcal{X} \cup \left\{x_{us}^j\right\}$
Output: $\mathcal{X}$

Algorithm 2: Heuristic Task Offloading Scheduling.

Initialize: $\mathcal{X} = \emptyset$
Find $x_{kw}^i = \underset{x_{us}^j, j \in \mathcal{N}, s \in \mathcal{S}, u \in \mathcal{U}}{\arg\max} J^*\left(\left\{x_{us}^j\right\}\right)$
Set $\mathcal{X} \leftarrow \left\{x_{kw}^i\right\}$
if there exists $x_{us}^j \in \mathcal{X}$ such that $J^*\left(remove\left(\mathcal{X}, x_{us}^j\right)\right) > \left(1 + \frac{1}{p(n,\epsilon)}\right) J^*\left(\mathcal{X}\right)$ **then**
  Set $\mathcal{X} \leftarrow remove\left(\mathcal{X}, x_{us}^j\right)$
  Go back to step 4
else if there exists $x_{us}^j \in \mathcal{G} \backslash \mathcal{X}$ such that $J^*\Big(exchange\left(\mathcal{X}, x_{us}^j\right)\Big) > \left(1 + \frac{1}{p(n,\epsilon)}\right) J^*\left(\mathcal{X}\right)$ **then**
  Set $\mathcal{X} \leftarrow exchange\left(\mathcal{X}, x_{us}^j\right)$
  Go back to step 4
end if
Output: $\mathcal{X}$

Remark 1: (Complexity Analysis of Algorithm 2) In Algorithm 2, $p(n, \epsilon)$ is a suitably chosen polynomial in n and $1/\epsilon$. Let INT be the initial value of problem (30) obtained at $\text{step}\,3$ in Algorithm 2, and OPT be the optimal value of the problem. The algorithm insists that each local step increases the value of the current solution by a factor of at least $\left(1 + \frac{1}{p(n,\epsilon)}\right)$. Let t be the number of local steps taken by Algorithm 2. It is clear that $\left(1 + \frac{1}{p(n,\epsilon)}\right)^t \leq \frac{OPT}{INT}$; and thus, $t = \mathcal{O}\left(p\left(n, \epsilon\right) \log \frac{OPT}{INT}\right)$. Note that the number of queries needed to calculate the value of the objective function in each iteration is at most n. In each query, one needs to solve an UPA problem and a CRA problem. Since the CRA problem has a closed-form solution, the running time complexity in each query mainly comes from solving the UPA problem, which takes $\lceil \log_2\left(P_u/\xi\right) \rceil$ iterations. Therefore, the running time of Algorithm 2 is $\mathcal{O}\left(\lceil \log_2\left(P_u/\xi\right) \rceil\, p\left(n, \epsilon\right) \log \frac{OPT}{INT}\right)$ which is polynomial in n and $1/\epsilon$. Furthermore, since the queries in

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

each iteration can be computed independently, the runtime can be greatly reduced by utilizing parallel computing.

Remark 2: (JTORA solution) Let $\mathcal{X}^*$ be the output of Algorithm 2. The corresponding solutions $\mathcal{P}^*$ for the uplink power allocation and $\mathcal{F}^*$ for computing resource sharing can be obtained using Algorithm 1 and the closed-form expression in (27), respectively, by setting $\mathcal{X} = \mathcal{X}^*$. Thus, the local optimal solution for the JTORA problem is $(\mathcal{X}^*, \mathcal{P}^*, \mathcal{F}^*)$. While characterizing the degree of suboptimality of the proposed solution is a non-trivial task—mostly due to the combinatorial nature of the task offloading decision and the nonconvexity of the original UPA problem—in the next section we will show via numerical results that our heuristic algorithm performs closely to the optimal solution using exhaustive search method.

VI. Performance Evaluation

Simulation results are presented to evaluate the performance of our proposed heuristic joint task offloading scheduling and resource allocation strategy, referred to as hJTORA. We consider a multi-cell cellular system consisting of multiple hexagonal cells with a BS in the center of each cell. The neighboring BSs are set 1 km apart from each other. We assume that both the users and BSs use a single antenna for uplink transmission and reception, respectively. The uplink channel gains are generated using a distance-dependent path-loss model given as $L\,[\mathrm{dB}] = 140.7 + 36.7\log_{10} d_{[\mathrm{km}]}$ [43], and the log-normal shadowing standard deviation is set to 8 dB. In most simulations, if not stated otherwise, we consider $S = 7$ cells and the users' maximum transmit power set to $P_u = 20$ dBm. In addition, the system bandwidth is set to $B = 20\,\mathrm{MHz}$ and the background noise variance is assumed to be $\sigma^2 = -100$ dBm.

In terms of computing resources, we assume the CPU capability of each MEC server and of each user to be $f_s = 20$ GHz and $f_u^l = 1$ GHz, respectively. According to the realistic measurements in [32], we set the energy coefficient κ as 5×10^{-27}. For computation task, we consider the face detection and recognition application for airport security and surveillance [4], which can highly benefit from the collaboration between mobile devices and MEC platform. Unless otherwise stated, we choose the default task input size as $d_u = 420$ KB (following [4], [10]), and the preference parameters as $\beta_u^t = 0.2$, $\beta_u^e = 0.8$, and $\lambda_u = 1$, $\forall u \in \mathcal{U}$. In addition, the users are dropped in random locations, with uniform distribution, within the coverage area of the network, and the number of sub-bands N is set equal to the number of users per cell. We compare the system utility performance of our proposed hJTORA strategy against the following baselines.

- *Exhaustive*: This is a brute-force method that finds the optimal offloading scheduling solution via exhaustive search over 2^n possible decisions; since the computational complexity of this method is very high, we only evaluate its performance in a small network setting.
- *Greedy Offloading and Joint Resource Allocation (**GOJRA**)*: All tasks (up to the maximum number that can be admitted by the BSs) are offloaded, as in [13]. In each cell, offloading users are *greedily* assigned to sub-bands that have the highest channel gains until all users are admitted or all the sub-bands are occupied; we then apply joint joint resource allocation across the BSs as proposed in Section V-A, B.
- *Independent Offloading and Joint Resource Allocation (**IOJRA**)*: Each user is randomly assigned a sub-band from its home BS, then the users independently make offloading decision [27]; joint resource allocation is employed.
- *Distributed Offloading and Resource Allocation (**DORA**)*: Each BS independently makes joint task offloading decisions and resource allocation for users within its cell [14].

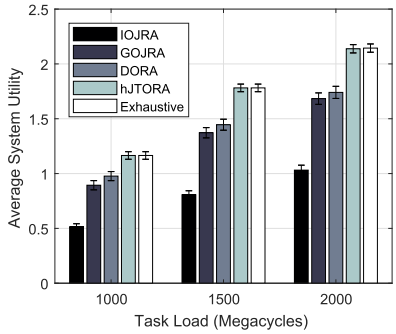


Fig. 2. Comparison of average system utility.

A. Suboptimality and Convergence Behavior of Algorithm 2

Firstly, to characterize the suboptimality of our proposed hJTORA solution obtained using Algorithm 2, we compare its performance with the optimal solution obtained by the *Exhaustive* method, and then with the three other described baselines. Since the *Exhaustive* method searches over all possible offloading scheduling decisions, its runtime is extremely long for a large number of variables; hence, we carry out the comparison in a small network setting with $U = 6$ users uniformly placed in the area covered by $S = 4$ cells, each having $N = 2$ sub-bands. We randomly generate 500 large-scale fading (shadowing) realizations and the average system utilities, with 95% Confidence Interval (CI), of different schemes are reported in Fig. 2 when we set $c_u = 1000, 1500$, and $2000\,\mathrm{Megacycles}$, respectively. It can be seen that the proposed hJTORA algorithm performs very closely to that of the optimal *Exhaustive* algorithm while it significantly outperforms the other baselines. We also observe that the performance of all schemes increases with the task workload. In all cases, hJTORA achieves an average system utility within 1% of that of the *Exhaustive* algorithm, while providing average gains of 31%, 38%, and 91% over DORA, GOJRA, and IOJRA schemes, respectively.

In Table II, we report the average runtime per simulation drop of different algorithms, running on a Windows 7 desktop PC with 3.6 GHz quad-core CPU and 16 GB RAM. It can be seen that the *Exhaustive* method takes very long time, about $100\times$ longer than the hJTORA algorithm for such a small network. The DORA algorithm runs slightly faster than hJTORA, while IOJRA and GOJRA requires the lowest runtimes. Furthermore, in order to evaluate the runtime of Algorithm 2 when more BSs cooperate with each other, we consider three deployment

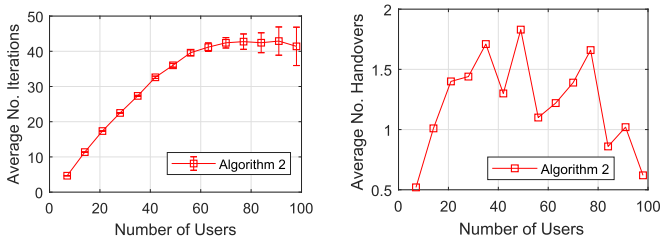
Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

TABLE II
RUNTIME COMPARISON

Algorithm	Runtime [ms]
IOJRA	0.20 ± 0.03
GOJRA	1.80 ± 0.2
DORA	6.80 ± 0.22
hJTORA	19.3 ± 0.7
Exhaustive	$1,923 \pm 1.4$

TABLE III
RUNTIME OF ALGORITHM 2 VERSUS NETWORK SIZE

Scenario	No. BSs	BS spacing	No. Users	Runtime [s]
Denser	9	237 m	18	0.59 ± 0.02
Very dense	16	209 m	32	3.29 ± 0.12
Ultra dense	25	112 m	50	12.46 ± 0.35



(a) (b)

Fig. 3. (a) Average number of iterations versus number of users. (b) Average number of handovers per offloading decision versus number of users; $c_u = 2000$ Megacycles.

scenarios: *denser*, *very dense*, and *ultra dense* networks for which the BS spacing (distance between two neighboring BSs) are set to 237 m, 209 m, and 112 m according to [44]. The number of BSs and users are set as in Table III, and the average runtime of Algorithm 2 (with 95% CI) are reported therein. We can see that the runtime increases considerably when there are many BSs cooperating with each other (e.g., 16 or 25). Therefore, in order to make the proposed solution practical, it is advisable to divide the network region into groups of cooperating BSs, each with fewer than 10 BSs.

B. Effect of Number of Users

In Fig. 3(a), we evaluate the convergence behavior of Algorithm 2 against different number of users by showing the average number of iterations with 95% CI. It can be seen that with a small and moderate number of users the number of iterations grows linearly (but less than) the number of users and the variation across different simulation runs is small. When the number of users is high, e.g., greater than 60, there is greater variation in the number of iterations but the growing rate of the average number of iteration gets lower. This shows that Algorithm 2 can scale well with the number of users. In Fig. 3(b), we plot the average number of handovers per simulation drop. While it can be seen that the average number of handovers (over 500 drops) varies with different number of users, it does not appear to follow a consistent trend.

We now evaluate the system utility performance against different number of users wishing to offload their tasks, as shown in Fig. 4(a, b, c). In particular, we vary the number of users per cell from 1 to 10 and perform the comparison in three scenarios with different task workloads. Note that the number of sub-bands N is set equal to the number of users per cell, thus the bandwidth allocated for each user decreases when there are more users in the system. Observe from Fig. 4(a, b, c) that hJTORA always performs the best, and that the performance of all schemes significantly increases when the tasks' workload increases. This is because when the tasks require more computation resources, the users will benefit more from offloading their tasks to the MEC servers. We also observe that, when the number of users is small, the system utility increases with the number of users; however, when the number of users exceeds some thresholds, the system utility starts to decrease. This is because, when there are many users competing for radio and computing resources for offloading their tasks, the overheads of sending the tasks and executing them at the MEC servers will be higher, thus degrading the offloading utility.

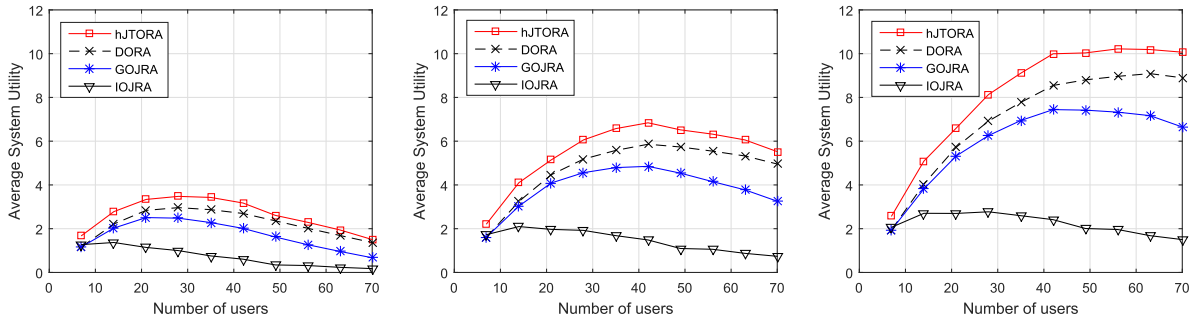
C. Effect of Task Profile

Here, we evaluate the system utility performance w.r.t. the computation tasks' profiles in terms of input size d_u's and workload c_u's. We consider two configuration of the MEC servers: (i) *homogeneous servers*—where all servers have the same CPU speed of 20 GHz, and (ii) *heterogeneous servers*—where the servers' CPU speeds are randomly selected from $\{10, 20, 30\}$ GHz. The average system utility of the four competing schemes are plotted in Fig. 5(a, b) with different values of c_u; and in Fig. 6(a, b) with different values of d_u. It can be seen that the average system utilities of all schemes increase with the task workload and decrease with the task input size. This implies that the tasks with small input sizes and high workloads benefit more from offloading than those with large input sizes and low workloads do. Moreover, we observe that the performance gains of the proposed hJTORA scheme over the baselines also follow the similar trend, i.e., they increase with task workloads and decrease with task input size. Notice that the difference in performance of all schemes in the homogeneous server setting versus in the heterogeneous server setting is marginal.

D. Effect of Users' Preferences

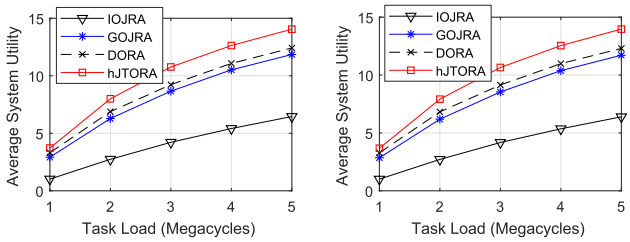
Figure 7(a, b) show the average time and energy consumption of all the users when we vary the users' preference to time, β_u^t's, from 0.1 to 0.9 while changing the users' preference to energy accordingly as $\beta_u^e = 1 - \beta_u^t, \forall u \in \mathcal{U}$. It can be seen that the average time consumption decreases when β_u^t increases, at the cost of higher energy consumption. In addition, the users experience a larger average time and energy consumption when there are more users in the system. This is because when there are more users competing for the limited resources, the chance that a user can benefit from offloading its task is lower.

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.



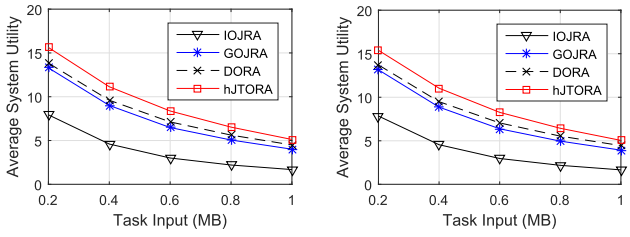
(a) $c_u = 1000$ Megacycles (b) $c_u = 1500$ Megacycles (c) $c_u = 2000$ Megacycles

Fig. 4. Comparison of average system utility against different number of users, evaluated on three different task workloads. (a) $c_u = 1000\,\text{Megacycles}$. (b) $c_u = 1500\,\text{Megacycles}$. (c) $c_u = 2000\,\text{Megacycles}, \forall \text{u} \in \mathcal{U}$.



(a) Homogeneous Servers (b) Heterogeneous Servers

Fig. 5. Comparison of average system utility against different task workloads, with $U = 28$ and $d_u = 420\,\text{KB}$.



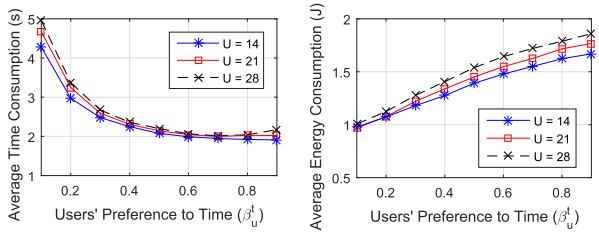
(a) Homogeneous Servers (b) Heterogeneous Servers

Fig. 6. Comparison of average system utility against different task input size, with $U = 28$ and $c_u = 3000\,\text{Megacycles}$.

E. Effect of Inter-Cell Interference Approximation

To test the effect of the approximation to model the inter-cell interference as in (21) in Section V-A, we compare the results of the hJTORA solution to calculate the system utility using the approximated expression versus using the exact expression of the inter-cell interference. Fig. 8 shows the system utility when the users' maximum transmit power P_u's vary between 0 and 35 dBm. It can be seen that the performance obtained using the approximated interference is almost identical to that of the exact interference when P_u is below 25 dBm, while an increasing gap appears when $P_u > 25$ dBm. However, as specified in the LTE standard, 3GPP TS36.101 section 6.2.3,[2] the maximum UE transmit power is 23 dBm; hence, we can argue that our approximation can work well in practical systems.

[2]Refer to: 3GPP TS36.101, V14.3.0, Mar. 2017



(a) (b)

Fig. 7. (a) Average time consumption and (b) energy consumption of all users obtained using hJTORA; with $c_u = 2000\,\text{Megacycles}, \forall \text{u} \in \mathcal{U}$.

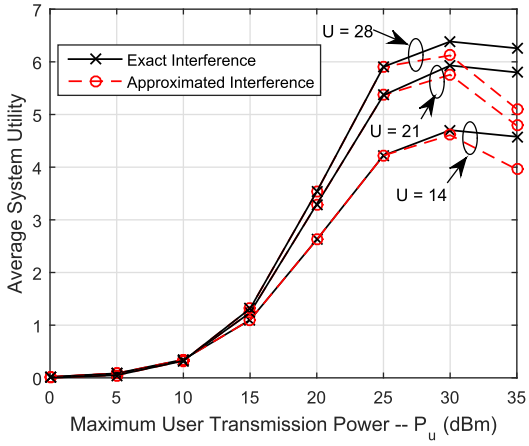


Fig. 8. Average system utility obtained by hJTORA solution with exact expression and approximation of the inter-cell interference; with $c_u = 1000\,\text{Megacycles}, \forall \text{u} \in \mathcal{U}$.

VII. Conclusions

We proposed a holistic strategy for a joint task offloading and resource allocation in a multi-cell Mobile-Edge Computing (MEC) network. The underlying optimization problem was formulated as a Mixed-Integer Non-linear Program (MINLP), which is very difficult to solve to optimal. Our approach decomposes the original problem into a Resource Allocation (RA) problem with fixed task offloading decision and a Task Offloading (TO) problem that optimizes the optimal-value function corresponding to the RA problem. We further decouple the RA problem into two independent subproblems, namely the

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

uplink power allocation and the computing resource allocation, and address them using quasi-convex and convex optimization techniques, respectively. Finally, we proposed a novel heuristic algorithm that achieves a suboptimal solution for the TO problem in polynomial time. Simulation results showed that our heuristic algorithm performs closely to the optimal solution and significantly improves the average system offloading utility over traditional approaches.

To further reduce the runtime of the proposed hJTORA algorithm, two approaches can be exploited: (i) *parallel computing*—since the queries in each iteration of Algorithm 2 are independent, they can be run in parallel to reduce the runtime in each iteration; (ii) *pre-disassociation*—for each user, the task offloading variables corresponding to the BSs that are far away can be set to zeros, thus reducing the search space of the task offloading decision before running Algorithm 2. In addition, as future work, it is worth characterizing the approximation ratio of the proposed algorithm with regard to the approximation of the interference term used in the uplink power allocation problem. Furthermore, it is also desirable to design a mistreatment-free solution where each user cannot improve its own performance by not following the network-wide solution.

APPENDIX A

Proof for Lemma 1: Firstly, it is straightforward to verify that $\Gamma_s(p_u)$ is twice differentiable on $\mathbb{R}$. We now check the second-order condition of a strictly quasi-convex function, which requires that a point p satisfying $\Gamma_s'(p)=0$ also satisfies $\Gamma_s'(p)>0$ [40]. The first-order and second-order derivatives of $\Gamma_s(p_u)$ can be calculated, respectively, as,

$$\Gamma_s'(p_u) = \frac{\psi_u C_u(p_u) - \frac{\vartheta_{us} D_u(p_u)}{A_u(p_u)\ln 2}}{C_u^2(p_u)}, \tag{31}$$

$$\Gamma_s''(p_u) = \frac{\vartheta_{us}\left[G_{us}(p_u) C_{us}(p_u) + 2\vartheta_{us} D_{us}(p_u)/\ln 2\right]}{A_{us}^2(p_u) C_{us}^3(p_u)\ln 2}, \tag{32}$$

in which,

$$A_{us}(p_u) = 1+\vartheta_{us} p_u, \tag{33a}$$

$$C_{us}(p_u) = \log_2(1+\vartheta_{us} p_u), \tag{33b}$$

$$D_{us}(p_u) = \phi_u + \psi_u p_u, \tag{33c}$$

$$G_{us}(p_u) = \vartheta_{us} D_{us}(p_u) - 2\psi_u A_{us}(p_u). \tag{33d}$$

Suppose that $\bar{p}_u \in (0, P_u]$; to satisfy $\Gamma_s'(\bar{p}_u)=0$, we need,

$$\Omega_s(\bar{p}_u) = \psi_u \log_2(1+\vartheta_{us}\bar{p}_u) - \frac{\vartheta_{us}(\phi_u+\psi_u\bar{p}_u)}{(1+\vartheta_{us}\bar{p}_u)\ln 2} = 0. \tag{34}$$

By substituting $\bar{p}_u$ into (32), we obtain,

$$\Gamma_s''(\bar{p}_u) = \frac{\vartheta_{us}^3 D_{us}^2(\bar{p}_u)}{A_{us}^2(\bar{p}_u) C_{us}^3(\bar{p}_u)\psi_u \ln^2 2}. \tag{35}$$

It can be easily verified that both ϑ_{us} and $D_{us}^2(\bar{p}_u)$ are strictly positive $\forall \bar{p}_u \in (0, P_u]$. Hence, $\Gamma_s''(\bar{p}_u) > 0$, which confirms that $\Gamma_s(p_u)$ is a strictly quasi-convex function in $(0, P_u]$.

APPENDIX B

Proof for Lemma 2: The Lagrangian of problem (25) can be calculated as,

$$\mathcal{L}(\Lambda(\mathcal{X},\mathcal{F}),\nu) = \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}_s}\frac{\eta_u}{f_{us}} + \sum_{s\in\mathcal{S}}\nu_s\left(\sum_{u\in\mathcal{U}} f_{us} - f_s\right), \tag{36}$$

where $\boldsymbol{\nu} = [v_1, ...v_S]$ is the vector of Lagrangian multipliers. Taking the derivatives of the Lagrangian w.r.t. f_{us}'s, we get,

$$\frac{\partial\mathcal{L}(\Lambda(\mathcal{X},\mathcal{F}),\nu)}{\partial f_{us}} = -\frac{\eta_u}{f_{us}^2} + \nu_s, \forall s\in\mathcal{S}, u\in\mathcal{U}_s. \tag{37}$$

By equating the gradient of the Lagrangian to zero and solving for f_{us}, the optimal computing resource allocation solution for problem (25) is obtained as,

$$f_{us}^* = \sqrt{\eta_u/\nu_s^*}, \forall s\in\mathcal{S}, u\in\mathcal{U}_s, \tag{38}$$

in which $\nu_s^* > 0$ is the constant satisfying,

$$\sum_{u\in\mathcal{U}} f_{us}^* = f_s, \forall s\in\mathcal{S}. \tag{39}$$

By substituting (38) into (39) and noting that $f_{us}^* = 0,\ \forall u\notin\mathcal{U}_s$, we obtain the optimal Lagrangian multiplier ν_s^* as,

$$\nu_s^* = \left(\frac{1}{f_s}\sum_{u\in\mathcal{U}_s}\sqrt{\eta_u}\right)^2, \forall s\in\mathcal{S} \tag{40}$$

Finally, by substituting (40) into (38), we can obtain the optimal solution to problem (25) in closed form as,

$$f_{us}^* = \frac{f_s\sqrt{\eta_u}}{\sum_{u\in\mathcal{U}_s}\sqrt{\eta_u}}, \forall s\in\mathcal{S}, u\in\mathcal{U}_s, \tag{41}$$

and the optimal objective function of problem (25) is then calculated as,

$$\Lambda(\mathcal{X},\mathcal{F}^*) = \sum_{s\in\mathcal{S}}\frac{1}{f_s}\left(\sum_{u\in\mathcal{U}_s}\sqrt{\eta_u}\right)^2. \tag{42}$$

REFERENCES

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI White Paper*, vol. 11, 2015.

[2] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[3] T. X. Tran, M.-P. Hosseini, and D. Pompili, "Mobile edge computing: Recent efforts and five key research directions," *IEEE COMSOC MMTC Commun.-Frontiers*, vol. 12, no. 4, pp. 29–33, 2017.

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

[5] K. Bilal, A. Erbad, and M. Hefeeda, "Crowdsourced multi-view live video streaming using cloud computing," *IEEE Access*, vol. 5, pp. 12635–12647, 2017.

[6] M.-P. Hosseini, T. X. Tran, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Deep learning with edge computing for localization of epileptogenicity using multimodal rs-fMRI and EEG big data," in *Proc. IEEE Int. Conf. Autonomic Comput.*, 2017, pp. 83–92.

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.

[7] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.
[8] V. Cardellini *et al.*, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, vol. 157, no. 2, pp. 421–449, 2016.
[9] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[10] 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.
[11] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 23–32, 2013.
[12] M. R. Rahimi, N. Venkatasubramanian, and A. V. Vasilakos, "Music: Mobility-aware optimal service allocation in mobile cloud computing," in *Proc. IEEE Int. Conf. Cloud Comput.*, 2013, pp. 75–82.
[13] 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, Jun. 2015.
[14] X. Lyu, H. Tian, P. Zhang, and C. Sengul, "Multi-user joint task offloading and resources optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.
[15] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.
[16] D. Lee *et al.*, "Coordinated multipoint transmission and reception in LTE-advanced: Deployment scenarios and operational challenges," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, Feb. 2012.
[17] T. X. Tran and D. Pompili, "Dynamic radio cooperation for user-centric cloud-RAN with computing resource sharing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2379–2393, Apr. 2017.
[18] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.
[19] Intel and Nokia Siemens Networks, "Increasing mobile operators' value proposition with edge computing," Technical Brief, 2013.
[20] Saguna and Intel, "Using mobile edge computing to improve mobile network performance and profitability," White paper, 2016.
[21] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. IEEE/IFIP Conf. Wireless On-Demand Netw. Syst. Services*, Feb. 2017, pp. 165–172.
[22] J. O. Fajardo, I. Taboada, and F. Liberal, "Improving content delivery efficiency through multi-layer mobile edge adaptation," *IEEE Netw.*, vol. 29, no. 6, pp. 40–46, Nov./Dec. 2015.
[23] Nokia, "LTE and Car2x: Connected cars on the way to 5G," 2016. [Online]: http://www.cambridgewireless.co.uk/Presentation/MB06.04.16-Nokia-UwePutzschler.pdf
[24] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 4, pp. 2322–2358, Sep.–Dec. 2017.
[25] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 369–392, Jan.–Apr. 2014.
[26] W. Zhang, Y. Wen, and D. O. Wu, "Energy-efficient scheduling policy for collaborative execution in mobile cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2013, pp. 190–194.
[27] 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.
[28] 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, Mar. 2015.
[29] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[30] 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.
[31] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2017, pp. 1–9.
[32] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, Jun. 2010, pp. 1–7.
[33] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2716–2720.
[34] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. New York, NY, USA: Academic, 2013.
[35] W. Saad, Z. Han, R. Zheng, M. Debbah, and H. V. Poor, "A college admissions game for uplink user association in wireless small cell networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2014, pp. 1096–1104.
[36] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*. Berlin, Germany: Springer Science & Business Media, 2006.
[37] T. X. Tran, N. H. Tran, H. R. Bahrami, and S. Sastry, "On achievable rate and ergodic capacity of NAF multi-relay networks with CSI," *IEEE Trans. Commun.*, vol. 62, no. 5, pp. 1490–1502, May 2014.
[38] K. Tammer, "The application of parametric optimization and imbedding to the foundation and realization of a generalized primal decomposition approach," *Math. Res.*, vol. 35, pp. 376–386, 1987.
[39] Y. Du and G. De Veciana, "Wireless networks without edges: Dynamic radio resource clustering and user scheduling," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2014, pp. 1321–1329.
[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[41] B. Bereanu, "Quasi-convexity, strictly quasi-convexity and pseudo-convexity of composite objective functions," *Revue Française D'automatique, Informatique, Recherche Opérationnelle Mathématique*, vol. 6, no. 1, pp. 15–26, 1972.
[42] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko, "Non-monotone submodular maximization under matroid and knapsack constraints," in *Proc. Annu. ACM Symp. Theory Comput.*, 2009, pp. 323–332.
[43] X. Chu, D. López-Pérez, Y. Yang, and F. Gunnarsson, *Heterogeneous Cellular Networks: Theory, Simulation and Deployment*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
[44] Nokia White Paper, "Ultra dense network (UDN)," 2016. [Online]: https://onestore.nokia.com/asset/200295/Nokia_Ultra_Dense_Network_White_Paper_EN.pdf

Tuyen X. Tran received the M.Sc. degree in electrical and computer engineering from the University of Akron, OH, USA, in 2013, and the B.Eng. degree (Hons.) in electronics and telecommunications from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2011. He received the Ph.D. degree in electrical and computer engineering from Rutgers University, NJ, USA, in May 2018. His research interests include the area of wireless communications, mobile cloud computing, and network optimization. He is currently a Senior Inventive Scientist at AT&T Labs Research, Bedminster, NJ. During the summers of 2015–2017, he held research internships at Huawei Technologies R&D Center, Bridgewater, NJ. He received the Best Paper Award at the IEEE/IFIP Wireless On-demand Network Systems and Services Conference in 2017 and the Outstanding Graduate Student Award from the Rutgers School of Engineering in 2018.

Dario Pompili received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2007. He had previously received the "Laurea" (combined B.S. and M.S. degrees) and Doctorate degrees in telecommunications and system engineering from the University of Rome "La Sapienza," Italy, in 2001 and 2004, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Rutgers U., where he directs the Cyber-Physical Systems Laboratory, which focuses on mobile computing, wireless communications and networking, acoustic communications, sensor networks, and datacenter management. He is a recipient of the NSF CAREER'11, ONR Young Investigator Program'12, and DARPA Young Faculty'12 Awards. In 2015, he was nominated Rutgers-New Brunswick Chancellor's Scholar. He published more than 150 refereed scholar publications: with more than 8400 citations, he has an h-index of 34 and a i10-index of 72 (Google Scholar, September 2018). He is a Senior Member of the IEEE Communications Society and ACM.

Authorized licensed use limited to: Yanshan University. Downloaded on June 15,2022 at 01:40:05 UTC from IEEE Xplore. Restrictions apply.