Cost-Effective User Allocation in 5G NOMA-based Mobile Edge Computing Systems

Phu Lai, Qiang He, *Senior Member, IEEE*, Guangming Cui, Feifei Chen, *Member, IEEE*, John Grundy, *Senior Member, IEEE*, Mohamed Abdelrazek, John Hosking, and Yun Yang, *Senior Member, IEEE*

Abstract—Mobile edge computing (MEC) allows edge servers to be deployed at cellular base stations. Mobile app vendors like Uber and YouTube can hire computing resources and deploy applications on edge servers for their users to access with a low latency connection. Non-orthogonal multiple access (NOMA) has emerged as a new technology with a high potential to support the massive connectivity of 5G networks, further enhancing the capability of MEC. The edge user allocation (EUA) problem, in which a mobile app vendor needs to allocate its users to edge servers to achieve certain optimization objectives, faces new challenges in 5G NOMA-based MEC systems. In this paper, we investigate the EUA problem in a multi-cell multi-channel downlink power-domain NOMA-based MEC system. The main objective is to help mobile app vendors maximize their benefit by allocating as many users as possible in a specific area at the lowest computing resource cost. To this end, we propose a decentralized game-theoretic approach to effectively select a channel and edge server for each user while fulfilling their resource requirements, e.g., CPU, RAM, storage, and data rate requirement. We theoretically and experimentally evaluate our solution, which significantly outperforms baseline and state-of-the-art approaches.

Index Terms—Mobile edge computing, non-orthogonal multiple access (NOMA), user allocation, power allocation, game theory

1 Introduction

Mobile edge computing (MEC) [1] is introduced to tackle one of the most challenging obstacles in cloud computing – high and unpredictable latency. By deploying edge servers at cellular base stations (BSs), mobile network operators can offer computing resources at the network edge, much closer to mobile users (referred to as "users" hereafter for simplicity). Mobile app vendors like Uber and YouTube can hire these computing resources for hosting their applications to serve their users with low latency. This is particularly critical for latency-sensitive applications and services such as facial recognition, interactive VR/AR gaming, vital monitoring systems, etc.

The rapid growth of mobile subscriptions promoted by 4G and the forthcoming 5G, which is predicted to reach 9 billions in 2025 [2], has put a great burden on the existing wireless communication infrastructure. Several multiple access techniques for wireless communication have been widely adopted for decades, e.g., code division multiple access (CDMA), time division multiple access (TDMA), and orthogonal frequency division multiple access (OFDMA). In conventional systems that employ those orthogonal multiple access (OMA) techniques, different users are allocated orthogonal resources in time, code, or frequency domain. Take OFDMA scheme for example, each individual user is allocated a dedicated channel, which prevents multiple access interference. However, such schemes are not capable of supporting a massive number of simultaneous users. As a result, non-orthogonal multiple access (NOMA) was proposed to support the massive connectivity demanded by 5G [3], [4]. NOMA achieves high spectral efficiency by allowing multiple users to be served simultaneously using the same time and frequency resources (channels) in power or code domain [4].

In this work, we focus on downlink NOMA networks in power domain because downlink is important for applications that transmit a substantial amount of data to users, e.g., video streaming or interactive VR/AR applications. To deal with the *intra-cell interference* caused by multiple users sharing the same channel, successive interference cancellation (SIC), a multi-user signal separation technique, is applied at the receivers when decoding wireless signals. By multiplexing users in the power domain at the transmitters (BSs) and employing SIC at the receivers (users), NOMA has been demonstrated to significantly improve the capacity and user throughput performance compared to conventional multiple access schemes [4]. NOMA is undoubtedly a promising enabler for 5G networks and has attracted tremendous attention from both academia and industry.

In addition to communication resources, computing resources hired on edge servers also need to be optimized. Similar to cloud computing, MEC also benefits from *multi-tenancy* [5], where multiple tenants/users can be simultaneously served by a single software instance or share the same infrastructure or database in an efficient manner [6], [7]. It allows higher resource utilization, energy efficiency, and overall performance on edge servers through workload consolidation [8]. In an MEC environment, multi-tenancy benefit can be achieved by allocating as many users as possible to an edge server as long as it does not overload the communication resources on the edge server with the

- *P. Lai, Q. He, G. Cui and Y. Yang are with the School of Software and Electrical Engineering, Swinburne University of Technology, 3122, Australia. E-mail: tlai, qhe, gcui, yyang@swin.edu.au.*
- *F. Chen and M. Abdelrazek are with the School of Information Technology, Deakin University, 3125, Australia. E-mail: feifei.chen, mohamed.abdelrazek@deakin.edu.au.*
- *J. Hosking is with the School of Science, University of Auckland, Auckland, New Zealand. E-mail: j.hosking@auckland.ac.nz.*
- *J. Grundy is with the Falcuty of Information Technology, Monash University, 3168, Australia. E-mail: john.grundy@monash.edu.*

incurred intra-cell interference. Leveraging multi-tenancy effectively allows an app vendor to reduce the amount of computing resources required to serve its users, saving system costs or operating costs. At the same time, saving edge servers' computing resources enables the app vendor to accommodate more users. This is essential for every app vendor and must be seriously considered in the allocation of users to edge servers.

The problem of allocating users to edge servers in an MEC system is referred to as an *edge user allocation* (EUA) problem. Recently, researchers are starting to investigate the impact of NOMA on the computation offloading problem [9] in MEC systems but not the EUA problem. Existing user allocation approaches do not consider both communication and computation aspects in the MEC system at the same time. User allocation approaches in pure cellular systems [10], [11], [12], [13] often lack the computation aspects of the scarcity and heterogeneity of computing resources on edge servers. Meanwhile, user allocation approaches in MEC systems [5], [14], [15], [16] often neglect the communication aspects of multiple wireless channels, interference, and power control, especially in a NOMA setting.

In this paper, from the app vendor's perspective, we make the first attempt to tackle the cost-effective EUA problem in a downlink multi-cell multi-user 5G NOMA-based MEC system. The app vendor needs to allocate each user by jointly making two different decisions: 1) user allocation, including edge server assignment and channel assignment, and 2) power allocation, such that the number of users allocated to edge servers is maximized, and the system cost (costs of computing resources, i.e., the amount of computing resources needed) is minimized. Meanwhile, a number of unique constraints of MEC systems must be satisfied (minimum user data rate requirement, proximity, and resource constraints, etc.), taking into account intra-cell and inter-cell interferences. Due to the $\mathcal{NP}$-hardness of this *NOMA-EUA* problem (to be proved later in this paper), it is intractable to find an optimal solution in large-scale scenarios. To tackle this challenge, we propose an efficient game-theoretic approach that can find NOMA-EUA solutions in a decentralized manner. The main contributions of this paper include:

- We formulate the NOMA-EUA problem, taking into account multiple channels, interferences, and power control, and prove its $\mathcal{NP}$-hardness.
- We formulate this problem as a potential game and theoretically analyze the existence of at least one Nash equilibrium in the game.
- To find the NOMA-EUA solution, we propose an iterative and decentralized algorithm, named miUA, for finding the Nash equilibrium in the game.
- We conduct a series of experiments to evaluate the performance of miUA. It is shown that miUA significantly outperforms all state-of-the-art and baseline approaches.

The remainder of the paper is organized as follows. Section 2 discusses the key motivation for this work and Section 3 introduces our NOMA system model. Section 4 formulates the NOMA-EUA problem, which consists of two subproblems: a user allocation problem and a power allocation problem. Section 5 proposes a solution to the user allocation problem, including a theoretical analysis. Section 6 shows a solution to the power allocation problem. miUA is experimentally evaluated in Section 7. In Section 8, we review the relevant literature. Finally, we conclude the paper and point out future work in Section 9.

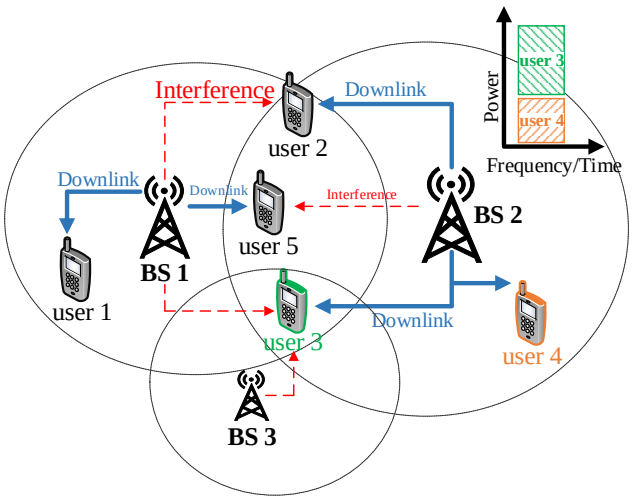


Fig. 1: An example of downlink multi-cell NOMA-based MEC networks

2 Motivation

In a 5G MEC system, BSs are densely distributed, especially in high-traffic areas. Their cell coverage areas usually partially overlap to minimize non-service areas – areas in which users can not be served by any edge server/BS[1]. A user located in the overlapping region can be allocated to one of the edge servers covering the user (*proximity constraint*) as long as that edge server has sufficient computing resources such as CPU or RAM, and communication resources such as channels (*resource constraint*). An edge server can serve users on a number of channels, where multiple users can be assigned to each single channel at the same time. Compared to a typical cloud environment, an edge server has a very limited amount of computing resources [17], [18]. Furthermore, a channel cannot be used to serve too many users at once due to the high interference it would cause. Thus, an ineffective user-to-edge-server allocation will soon exhaust the computing and networking resources and result in poor data rate for users. Similar to [5], [19], [20], [21], we consider a quasi-static scenario where users are relatively stationary and do not quickly travel across different BSs/edge servers when they are being allocated to edge servers, e.g. mobile or IoT users who are not moving at a high speed, traffic cameras, smart sensors, etc.

In downlink NOMA, SIC is facilitated by differentiating the transmit power between users sharing the same channel [3]. In single-cell NOMA scenarios, to ensure successful decoding of the superposed signal sent by a BS, stronger users on a channel (who have higher channel gains) are

1. We speak interchangeably of edge servers and base stations. For the sake of consistency, we will hereafter try to use the term "edge server" instead of "base station". In situations where the communication/networking aspects are discussed, "base station" will be used.

allocated less transmit power, and weaker users (who have lower channel gains) are allocated more transmit power [22]. Each user employs SIC to remove the signal interference caused by users with lower channel gains. Nevertheless, decoding solely by the order of channel gains is not applicable to multi-cell NOMA scenarios because users' channel conditions are now also affected by the *inter-cell interference* caused by their unassociated neighbor BSs [12], which could be very severe in a dense multi-cell network.

Take Fig. 1 for example, where user 3 and user 4 suffer from intra-cell interference as they share the same channel on the same edge server. In addition to intra-cell interference, user 3 is also impacted by the inter-cell interference caused by its neighbor BSs, i.e., BS 1 and BS 2. Fortunately, it has been demonstrated that an effective power allocation and decoding order can considerably reduce inter-cell interference [12], which in turn improves users' data rate, or system throughput in general [12], [22]. Thus, the EUA problem must be jointly solved with the power allocation problem to ensure that every user receives an app-specific satisfactory data rate.

3 System Model

3.1 System Description

Edge Servers: An MEC system consists of a set of M BSs denoted by $\mathcal{S} = \{s_1, s_2, ..., s_M\}$. Each BS is equipped with an edge server. Each edge server $s_j \in \mathcal{S}$, $j \in \{1, 2, ..., M\}$, has a specific amount of computing resources of different types $\mathcal{T} = \{\text{CPU}, \text{RAM}, \text{storage}, ...\}$. The computing capacity of an edge server s_j is represented by a $|\mathcal{T}|$-dimensional vector $Q_j = (Q_j^t)$, where each dimension Q_j^t is the capacity of resource type $t \in \mathcal{T}$. Each edge server s_j covers a specific geographic area with cell radius R_j.

For each edge server s_j, the total bandwidth B is equally divided into a set of V channels denoted by $\mathcal{C}_j = \{c_j^1, c_j^2, ..., c_j^V\}$. The bandwidth of each channel $c_j^k \in \mathcal{C}_j$ is thus $B_j^k = B/V$, where $k \in \{1, 2, ..., V\}$.

Mobile users: The set of all the N users is denoted by $\mathcal{U} = \{u_1, u_2, ..., u_N\}$. For each user $u_i \in \mathcal{U}$, $i \in \{1, 2, ..., N\}$, let a $|\mathcal{T}|$-dimensional vector $w_i = (w_i^t)$, $t \in \mathcal{T}$, denote user u_i's computing resource requirement, i.e., the amount of computing resources that could be consumed by an edge server assigned to serve user u_i. Let $d_{j,i}$ be the distance between user u_i and edge server s_j, and $\mathcal{S}_i = \{s_j \in \mathcal{S} | d_{j,i} \leq R_j\}, \forall u_i \in \mathcal{U}$, be the set of user u_i's neighbor edge servers, i.e., edge servers that have user u_i in their cell coverage areas. Note that an individual user can be allocated to only one channel on an edge server. To allocate each user $u_i \in \mathcal{U}$, two decisions need to be made as defined below:

Definition 1. ***(User Allocation Decision)*** *Given the set of edge server $\mathcal{S} = \{s_1, ..., s_M\}$, each edge server $s_j \in \mathcal{S}$ has a set of channels $\mathcal{C}_j = \{c_j^1, ..., c_j^V\}$, let $\mathbf{a}_{j,i}^k = \{0, 1\}$ be the binary decision variable for user u_i. We have $\mathbf{a}_{j,i}^k = 1$ if user u_i is allocated to edge server s_j on channel c_j^k; otherwise $\mathbf{a}_{j,i}^k = 0$. We use $\boldsymbol{a} = \{\boldsymbol{a}_i | u_i \in \mathcal{U}\}$ to denote the user allocation strategy composed by the decisions for all the users $\forall u_i \in \mathcal{U}$, and $\boldsymbol{a}_i \triangleq (s_j, c_j^k)$, where $\mathbf{a}_{j,i}^k = 1$, $s_j \in \mathcal{S}$, $c_j^k \in \mathcal{C}_j$, which indicates the channel and edge server to which user u_i is allocated.*

Let $\mathcal{U}_j = \{u_i \in \mathcal{U} | \sum_{k=1}^{V} \mathbf{a}_{j,i}^k = 1\}, \forall s_j \in \mathcal{S}$, denote the set of users allocated to edge server s_j, and $\mathcal{U}_j^k = \{u_i \in \mathcal{U} | \mathbf{a}_{j,i}^k = 1\}, \forall s_j \in \mathcal{S}, \forall c_j^k \in \mathcal{C}_j$, denote the set of users allocated to channel c_j^k on edge server s_j.

Definition 2. ***(Power Allocation Decision)*** *Let $\boldsymbol{p}_{j,i}^k$ denote the transmit power allocated to user u_i on channel c_j^k of edge server s_j, i.e., the amount of power used by edge server s_j to transmit data to user u_i on channel c_j^k. We use $\boldsymbol{p} = \{\boldsymbol{p}_{j,i}^k | u_i \in \mathcal{U}, s_j \in \mathcal{S}, c_j^k \in \mathcal{C}_j\}$ to denote the power allocation strategy composed by the power allocation decisions for all the users.*

The notations used in this paper are summarized in Appendix **??** of the supplementary file.

3.2 Signal Model

According to the NOMA scheme [4], edge server s_j broadcasts a superposition-coded signal x_j^k to all the users allocated on channel c_j^k simultaneously. The transmitted signal x_j^k can be expressed as follows:

$$x_j^k = \sum_{u_i \in \mathcal{U}_j^k} \sqrt{\mathbf{p}_{j,i}^k} x_{j,i}^k \tag{1}$$

where $x_{j,i}^k$ is the signal transmitted from edge server s_j to user u_i on channel c_j^k. NOMA facilitates a simultaneous transmission of multiple users' signals [12], [23], whose power levels are differentiated, over the same transmission period and channel. We denote the total transmit power of edge server s_j on channel c_j^k by $p_j^k = \sum_{u_i \in \mathcal{U}_j^k} \mathbf{p}_{j,i}^k$. The total transmit power allocated to all the users on all the channels of an edge server s_j must not exceed its maximum transmit power P_j: we have $\sum_{c_j^k \in \mathcal{C}_j} p_j^k \leq P_j$.

For each user u_i allocated to edge server s_i on channel c_j^k (when $\mathbf{a}_{j,i}^k = 1$), its received signal $y_{j,i}^k$ is the summation of its intended signal, intra-cell interference (caused by other users sharing the same channel), inter-cell interference (caused by nearby BSs/edge servers), and other noise. Note that (1) includes both the signal intended for user u_i and the signal intended for the other users sharing the same channel with u_i, which causes intra-cell interference. $y_{j,i}^k$ is defined as follows:

$$y_{j,i}^k = \underbrace{h_{j,i}^k x_j^k}_{\substack{\text{intended signal} \\ \text{+ intra-cell interference}}} + \underbrace{\sum_{s_l \in \mathcal{S} \setminus \{s_j\}} h_{i,l}^k x_l^k}_{\text{inter-cell interference}} + \underbrace{o_{j,i}^k}_{\text{noise}} \tag{2}$$

where $h_{j,i}^k$ is the complex channel coefficient between user u_i and edge server s_j on channel c_j^k, and $o_{j,i}^k$ is the additive white Gaussian noise with variance σ^2, i.e., $o_{j,i}^k \sim \mathcal{CN}(0, \sigma^2)$. User u_i's channel gain on channel c_j^k is $|h_{j,i}^k|^2$, which includes all the factors that can influence a signal.

3.3 Successive Interference Cancellation

In a downlink NOMA system, SIC is implemented for users sharing the same channel, i.e. $\mathcal{U}_j^k$, so that they can decode their received superposed signal. Assume that $\mathcal{U}_j^k$ has been determined, i.e., some users have been allocated to the k-th channel on edge server s_j. With SIC, users with stronger channel conditions detect and remove the signals of users with weaker channel conditions, who treat the signals of users with stronger channel conditions as noise [4]. Without loss of generality, suppose all the users in $\mathcal{U}_j^k$ are

ordered by their channel conditions: $u_1, u_2, ..., u_{|\mathcal{U}_j^k|}$, where u_1 has the weakest channel condition and $u_{|\mathcal{U}_j^k|}$ has the strongest channel condition. SIC is not required for user u_1 since it is the first in $\mathcal{U}_j^k$ to decode signal. User u_1 first decodes $x_{1,j}^k$ and subtracts its components from $y_{1,j}^k$. The user who comes next in the decoding order (user u_2) can thus decode its received signal without interference from user u_1. Following this principle, the signal received by user $u_i \in \mathcal{U}_j^k$ has the following signal-to-interference-plus-noise ratio (SINR):

$$\gamma_{j,i}^k = \frac{|h_{j,i}^k|^2 \mathbf{p}_{j,i}^k}{|h_{j,i}^k|^2 \sum_{q=i+1}^{|\mathcal{U}_j^k|} \mathbf{p}_{j,q}^k + I_{j,i}^k + \sigma^2} \tag{3}$$

where $I_{j,i}^k = \sum_{s_l \in \mathcal{S}_i \setminus \{s_j\}} |h_{l,i}^k|^2 p_l^k$ is the inter-cell interference caused by user u_i's neighbor edge servers on channel c_j^k. Given (3), the SINR of user $u_{|\mathcal{U}_j^k|}$, which is the last user to decode the received signal, is: $\gamma_{|\mathcal{U}_j^k|,j}^k = (|h_{|\mathcal{U}_j^k|,j}^k|^2 \mathbf{p}_{|\mathcal{U}_j^k|,j}^k)/(I_{|\mathcal{U}_j^k|,j}^k + \sigma^2)$.

Suppose $u_i, u_q \in \mathcal{U}_j^k$ and $i < q$, i.e., user u_q has a stronger channel condition. According to [12], [24], [25], in order to ensure a successful SIC, user u_q's achievable data rate for decoding user u_i's signal must be greater than or equal to user u_i's data rate for decoding its own signal: $r_{j,q\to i}^k \geq r_{j,i\to i}^k$. If this condition is not satisfied, user u_i's achievable data rate will decrease due to the intra-cell interference not being canceled. Thus, user u_i's achievable data rate $r_{j,i}^k$ on channel c_j^k can be given by:

$$r_{j,i}^k = \min\{r_{j,q\to i}^k | \forall q \geq i\} \tag{4}$$

where $r_{j,q\to i}^k$, i.e. user u_q's data rate for decoding user u_i's signal is:

$$r_{j,q\to i}^k = B_j^k \log_2\left(1 + \frac{|h_{j,q}^k|^2 \mathbf{p}_{j,i}^k}{|h_{j,q}^k|^2 \sum_{t=i+1}^{|\mathcal{U}_j^k|} \mathbf{p}_{j,t}^k + I_{j,q}^k + \sigma^2}\right) \tag{5}$$

Intuitively, user u_i's achievable data rate is the minimum data rate of the users that come after user u_i in the SIC decoding order, which will be discussed next.

SIC Decoding Order. As analyzed above, the position of a user in the decoding order plays an important role in its achievable data rate. Therefore, the decoding order cannot be overlooked when the data rate is being optimized. As can be seen from (4) and (5), the data rate is partially determined by the channel coefficient and inter-cell interference. By transforming (4), user u_i's achievable data rate $r_{j,i}^k$ can be expressed by:

$$r_{j,i}^k = B_j^k \log_2\left(1 + \frac{\mathbf{p}_{j,i}^k}{\sum_{t=i+1}^{|\mathcal{U}_j^k|} \mathbf{p}_{j,t}^k + H_{j,i}^k}\right) \tag{6}$$

where

$$H_{j,i}^k = \max\left\{\frac{I_{j,q}^k + \sigma^2}{|h_{j,q}^k|^2} \middle| \forall q \geq i\right\} \tag{7}$$

To ensure an acceptable data rate with low transmit power for all the users, the decoding order should be determined based on their channel conditions and the inter-cell interference as follows: $H_{j,1}^k \geq ... \geq H_{j,|\mathcal{U}_j^k|}^k$. This order is guaranteed if the decoding order of users allocated to channel c_j^k on edge server s_j follows the sequence:

$$\Theta(c_j^k) \triangleq \frac{I_{j,1}^k + \sigma^2}{|h_{j,1}^k|^2} \geq ... \geq \frac{I_{j,|\mathcal{U}_j^k|}^k + \sigma^2}{|h_{j,|\mathcal{U}_j^k|}^k|^2} \tag{8}$$

It has been shown that this decoding order is an optimal order for efficiently increasing the data rate of each individual users [12]. If this decoding order is satisfied, user u_i's achievable data rate $r_{j,i}^k$ is:

$$r_{j,i}^k = B_j^k \log_2\left(1 + \frac{|h_{j,i}^k|^2 \mathbf{p}_{j,i}^k}{|h_{j,i}^k|^2 \sum_{t=i+1}^{|\mathcal{U}_j^k|} \mathbf{p}_{j,t}^k + I_{j,i}^k + \sigma^2}\right) \tag{9}$$

3.4 Resource Utilization Model

Multi-tenancy is an important feature in computing resource management [7] and must also be considered in EUA [5]. By allowing users to share the same software instance, app vendors can efficiently utilize the computing resources hired on edge servers. This is critical in MEC systems where computing resources on edge servers are relatively scarce [21]. This drives app vendors to aggregate their users to a small set of edge servers. For example, say two users need to be served who require one CPU each. Allocating them to two different edge servers would require two CPUs to serve them. Taking advantage of multi-tenancy, allocating them to the same edge server to be served by the same software instance would require slightly less than two CPUs. According to [5], [8], taking into account multi-tenancy, the CPU utilization of edge server s_j can be estimated based on the users served by s_j:

$$f_j^{\text{cpu}} = -\log_z(|\mathcal{U}_j|)/100 \tag{10}$$

where z is determined by the app-specific computation task size ($0.9 < z < 1$) and $|\mathcal{U}_j| > 1$ is the number of users served by edge server s_j. In general, when the number of users served by an edge server increases, the CPU utilization of that edge server increases monotonically until it converges at some point. The convergence occurs when the number of users is overly large and incurs a high computational overhead for resource sharing [8]. When the extra computational overhead exceeds the corresponding multi-tenancy benefit, it does not benefit the app vendor as much as before, and it is more cost-effective to serve the extra users with another edge server.

As shown in [8], the storage utilization also follows a model similar to (10). Assuming the utilization other computing resources, e.g., RAM or storage, also follows a similar model, the utilization of the computing resource $t \in \mathcal{T}$ on edge server s_j when user u_i is being allocated can be measured by:

$$f_{j,i}^t = -\log_{z^t}(|\mathcal{U}_j|)/100 \tag{11}$$

where z^t is determined by the computation task size and is dependent of computing resource type $t \in \mathcal{T}$, $|\mathcal{U}_j|$ is the number of users on server s_j to which user u_i is allocated. We have $0 < f_{j,i}^t < 1$, $\forall t \in \mathcal{T}$, $\forall s_j \in \mathcal{S}$. From the app vendor's perspective in this paper, the user allocation is app-specific. We assume that the computation task sizes of users with different resource requirements are identical and do not vary during and after the allocation process.

3.5 Computing Resource Cost Model

In an MEC system, an app vendor needs to pay for computing resources hired on edge servers to serve its users. Thus, it is important to utilize multi-tenancy to the fullest extent to save on computing resource costs. Given a user allocation strategy $\mathbf{a}$, the computing resource cost incurred by the decision of user $u_i \in \mathcal{U}$ is:

$$M_{\mathbf{a}}(u_i) = \begin{cases} \sum\limits_{t\in\mathcal{T}} \tau^t (1 - f_{j,i}^t) w_i^t, & \text{if } \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k = 1 \\ \epsilon \sum\limits_{t\in\mathcal{T}} \tau^t w_{max}^t, & \text{if } \sum\limits_{s_j\in\mathcal{S}} \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k = 0 \end{cases} \tag{12}$$

where $(1 - f_{j,i}^t) w_i^t$ is the required amount of computing resource of type $t \in \mathcal{T}$ on an edge server when $\sum_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k = 1$, i.e., user u_i is allocated to edge server s_j, and τ^t is the weight that indicates the app vendor's priority for saving computing resource of type $t \in \mathcal{T}$ by leveraging multi-tenancy. For example, if an app is compute-intensive, saving processing power such as CPU would be more beneficial than saving other computing resources such as storage. When $\sum_{s_j\in\mathcal{S}} \sum_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k = 0$, i.e., user u_i is not allocated to any edge server, the cost incurred is modeled as $\epsilon \sum_{t\in\mathcal{T}} \tau^t w_{max}^t$, where $\epsilon > 1$ is the weight that indicates the severity of the penalty when the user is unallocated, w_{max}^t is the maximum amount of computing resource of type $t \in \mathcal{T}$ that a user in the system may consume. From the app vendor's perspective, it is critical to allocate a user to an edge server to ensure its low-latency service. Thus, the cost incurred by failing to allocate a user is always greater than $\sum_{t\in\mathcal{T}} \tau^t (1 - f_{j,i}^t) w_i^t$ - the cost incurred when the user is allocated to an edge server. This drives the app vendor to allocate as many users as possible to edge servers. Otherwise, it will just simply choose not to allocate any users to any edge servers to minimize the incurred system cost to zero.

4 Problem Formulation

In this section, we model the NOMA-EUA problem as a mixed-integer constrained optimization problem as follows:

$$\min_{\{\mathbf{a},\mathbf{p}\}} \sum_{i=1}^{N} M_{\mathbf{a}}(u_i) \tag{13a}$$

$$\text{s.t.} \sum_{i=1}^{N} \sum_{t=1}^{|\mathcal{T}|} \mathbf{a}_{j,i}^t (1 - f_{j,i}^t) w_i^t \le Q_j^t, \forall s_j \in \mathcal{S} \tag{13b}$$

$$\sum_{j=1}^{M} \sum_{k=1}^{V} \mathbf{a}_{j,i}^k d_{j,i} \le R_j, \forall u_i \in \mathcal{U} \tag{13c}$$

$$\sum_{j=1}^{M} \sum_{k=1}^{V} \mathbf{a}_{j,i}^k \le 1, \forall u_i \in \mathcal{U} \tag{13d}$$

$$\mathbf{a}_{j,i}^k r_{j,i}^k \ge \mathbf{a}_{j,i}^k \Upsilon, \forall s_j \in \mathcal{S}, \forall c_j^k \in \mathcal{C}_j, \forall u_i \in \mathcal{U} \tag{13e}$$

$$\Theta(c_j^k), \forall s_j \in \mathcal{S} \tag{13f}$$

$$\sum_{k=1}^{V} \sum_{i=1}^{N} \mathbf{a}_{j,i}^k \mathbf{p}_{j,i}^k \le P_j, \forall s_j \in \mathcal{S} \tag{13g}$$

$$\mathbf{a}_{j,i}^k \in \{0, 1\}, \forall s_j \in \mathcal{S}, \forall c_j^k \in \mathcal{C}_j, \forall u_i \in \mathcal{U} \tag{13h}$$

$$\mathbf{p}_{j,i}^k \in \mathbb{R}_{\ge 0}, \forall s_j \in \mathcal{S}, \forall c_j^k \in \mathcal{C}_j, \forall u_i \in \mathcal{U} \tag{13i}$$

where $\mathbf{a}$ is the user allocation strategy, and $\mathbf{p}$ is the power allocation strategy. Optimization objective (13a) minimizes the total system cost, i.e., the computing resource cost modeled in Section 3.5. Computing resource constraint (13b) ensures that the aggregated computing resource requirements of all the users allocated to an edge server must not exceed the computing capacity of that edge server. Proximity constraint (13c) ensures that an edge server can only serve users within its coverage area. Constraint (13d) indicates that any user can only be either unallocated, or be allocated to one channel of an edge server. Constraint (13e) ensures a minimum app-specific data rate Υ for each allocated user. Constraint (13f) enforces the optimal decoding order stated in Section 3.3, which allows any user to successfully decode the signals of users with weaker channel conditions on the same channel. Constraint (13g) ensures that the total transmit power of all users allocated to an edge server must not exceed its maximum power allowance. Constraints (13h) and (13i) indicate the possible values of user allocation decisions $\mathbf{a}_{j,i}^k$ and transmit power decisions $\mathbf{p}_{j,i}^k$.

The optimization problem above can be proved to be $\mathcal{NP}$-hard by showing that its subproblem (Section 4.1) is $\mathcal{NP}$-hard. Considering the dynamic of channel conditions associated with different edge servers, this NOMA-EUA problem becomes even more intractable to solve. To solve it efficiently, we decompose it into two subproblems: 1) a user allocation problem, and 2) a power allocation problem. The user allocation problem will be solved first to find a user-to-channel allocation that fully utilizes the computing resources on edge servers. Given the user allocation strategy, transmit power will be allocated to users such that they can achieve the minimum user data rate requirement.

4.1 User Allocation Problem

In this section, we formulate the user allocation problem. The power allocation problem is formulated in Section 4.2. The user allocation problem can be modeled as follows:

$$\begin{aligned} \min_{\{\mathbf{a}\}} \quad & C_{\mathbf{a},\mathbf{p}} \triangleq \sum_{i=1}^{N} \left(\eta_1 M_{\mathbf{a}}(u_i) + \eta_2 I_{\mathbf{a},\mathbf{p}}(u_i)\right) \\ \text{s.t.} \quad & \text{(13b), (13c), (13d), (13h)} \end{aligned} \tag{14}$$

where η_1 and η_2 ($\eta_1 + \eta_2 = 1$) are the weight parameters that indicates the relative importance of the two types of costs, $M_{\mathbf{a}}(u_i)$ and $I_{\mathbf{a},\mathbf{p}}(u_i)$. In optimization objective (14), we add a new term $I_{\mathbf{a},\mathbf{p}}(u_i)$, i.e., the interference cost. The objective of this subproblem is to minimize the cost $C_{\mathbf{a},\mathbf{p}}$ incurred by serving all the users, including computing resource cost $M_{\mathbf{a}}(u_i)$ and interference cost $I_{\mathbf{a},\mathbf{p}}(u_i)$. If the computing resource cost is the sole cost, (14) will pursue to allocate as many users to as few edge servers as possible. This will easily cause severe intra-cell and inter-cell interference in a NOMA-based MEC system, consequently reducing user data rate and increasing transmit power consumption. To mitigate this issue, we take the costs of interference into consideration. Given a user allocation strategy $\mathbf{a}$ and a power allocation strategy $\mathbf{p}$, the interference-plus-noise experienced by a user u_i, denoted by $I_{\mathbf{a},\mathbf{p}}(u_i)$, is defined

as:

$$I_{\mathbf{a},\mathbf{p}}(u_i) = \begin{cases} |h_{j,i}^k|^2 \sum_{q=i+1}^{|\mathcal{U}_j^k|} \mathbf{p}_{j,q}^k + I_{j,i}^k + \sigma^2, \text{if} \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k = 1 \\ \varepsilon I_{max}, \qquad\qquad \text{if} \sum\limits_{s_j \in \mathcal{S}} \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k = 0 \end{cases} \tag{15}$$

where $\varepsilon > 1$ is the weight specified by the app vendor that indicates the severity of the penalty when the user is unallocated, I_{max} is the maximum interference-plus-noise that a user could experience. It is formulated in this way so that the interference cost of unallocated users is always greater than the interference cost of allocated users, thus driving app vendors to allocate users to edge servers. The computing resource cost has also been formulated by following this methodology in Section 3.5. We can prove the $\mathcal{NP}$-hardness of this problem by reducing the $\mathcal{NP}$-complete PARTITION problem [26] to a special case of the decision version of this problem. The detailed proof can be found in Appendix **??**.

Note that power allocation is not considered in this subproblem so all the users are assumed to be allocated with identical and fixed transmit power for now. In the implementation, $M_{\mathbf{a}}(u_i)$ and $I_{\mathbf{a},\mathbf{p}}(u_i)$ will be min-max normalized. The possible minimum and maximum values of computing resource and interference costs can easily be calculated based on the given edge server information in real-world scenarios, i.e. edge server computing resource capacity, available channels, edge server locations, and minimum user data rate requirement.

Constraints related to power and data rate, including (13e), (13f), (13g), and (13i) are not considered in this subproblem because they do not contribute to the optimization of computing resources. These constraints will be enforced through power allocation.

4.2 Power Allocation Problem

The power allocation problem is to be solved next. It is expressed as follows:

$$\begin{aligned} &\min_{\{\mathbf{p}\}} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{V} \mathbf{p}_{j,i}^k \\ &\text{s.t. (13e), (13f), (13g), (13i)} \end{aligned} \tag{16}$$

The main objective of this subproblem is to allocate to users as little transmit power as possible while satisfying the user's minimum data rate requirement, SIC decoding order constraint, and power capacity constraint. Note that our main goal is to help app vendors solve the NOMA-EUA problem with the goal to minimize the system cost (computing resource cost). Here, we minimize the transmit power allocated to individual users to keep in line with the cost-saving initiatives. Another main reason is that a data rate higher than what is required for accessing an app is not necessary for most, if not all, apps. Nonetheless, other possible optimization objectives, e.g., maximizing users' overall data rate, can be pursued here instead of (16) without fundamentally modifying the problem model.

5 User Allocation

In this section, we present a game-theoretic approach employed by miUA to effectively and efficiently solve the user allocation problem introduced in Section 4.1. The power allocation problem introduced in Section 4.2 will be solved in Section 6. Over the years, game theory has been shown to be a versatile method for solving $\mathcal{NP}$-hard problems in MEC systems [5], [20], [21]. In this paper, players are simulated to make allocation decisions individually, pursuing to achieve objective (14). The game is decentralized by design and can alleviate the computational overhead that occurs by a centralized optimal solution.

5.1 Game Formulation and Properties

Our game aims to find a user allocation strategy $\mathbf{a}$, which consists of the allocation decisions for all the users. Those decisions are made to pursue the app vendor's objective (14) by following the rules of the game. Let $\mathbf{a}_{-i} = (\mathbf{a}_1, ..., \mathbf{a}_{i-1}, \mathbf{a}_{i+1}, ..., \mathbf{a}_N)$ denote the user allocation strategy except the decision for user u_i. Based on other users' decisions $\mathbf{a}_{-i}$, each individual user u_i will try to make a suitable decision on which channel of which edge server to be allocated to, so that the total system cost is minimized.

The user allocation problem is modeled as a game $\Gamma = (\mathcal{U}, \{\mathcal{A}_i\}_{u_i \in \mathcal{U}}, \{C_{\mathbf{a},\mathbf{p}}(\mathbf{a}_i)\}_{u_i \in \mathcal{U}})$, where $\mathcal{U}$ is the set of users (players), $\mathcal{A}_i$ is the set of possible allocation strategies for each user u_i, and $C_{\mathbf{a},\mathbf{p}}(\mathbf{a}_i)$ is the cost function that measures the cost incurred by user u_i's decision $\mathbf{a}_i = (s_j, c_j^k)$, the lower the better, $C_{\mathbf{a},\mathbf{p}}(\mathbf{a}_i) = \sum_{u_q \in \mathcal{U}_l \cup \mathcal{U}_j} (\eta_1 M_{\mathbf{a}}(u_q) + \eta_2 I_{\mathbf{a},\mathbf{p}}(u_q))$, where $\mathcal{U}_l$ is the set of users allocated to server s_l, which is the server to which user u_i was allocated (if any) before it is allocated to server s_j. Both $\mathcal{U}_l$ and $\mathcal{U}_j$ must be considered because switching a user from a server s_l to another server s_j impacts the inter-cell and inter-cell interference received by the users allocated to server s_l and s_j.

Next, we show that this game admits at least one Nash equilibrium – a stable state of the game where the system cost cannot be further lowered by changing the decision for any individual users unilaterally.

Definition 3. ***(Nash Equilibrium)*** *A user allocation strategy $\boldsymbol{a}^* = (\boldsymbol{a}_1^*, ..., \boldsymbol{a}_N^*)$ is a Nash equilibrium when no user can unilaterally change its decision to further lower the system cost:*

$$C_{\boldsymbol{a}_{-i}^*,\boldsymbol{p}}(\boldsymbol{a}_i^*) \leq C_{\boldsymbol{a}_{-i}^*,\boldsymbol{p}}(\boldsymbol{a}_i), \forall \boldsymbol{a}_i \in \mathcal{A}_i, \forall u_i \in \mathcal{U} \tag{17}$$

Lemma 1 guarantees that if there exists a Nash equilibrium, the decisions for all the users will finally constitute an allocation strategy that reaches the Nash equilibrium through finite iterations in the game.

Lemma 1. *Given a Nash equilibrium $\boldsymbol{a}^*$ of the game, the allocation decision $\boldsymbol{a}_i^* \in \mathcal{A}_i$ made for each user $u_i \in \mathcal{U}$ is the best response to the decisions $\boldsymbol{a}_{-i}$ made for the other $n-1$ users.*

Proof: Please refer to Appendix **??**. □

An essential property of a potential game is that it always admits at least one Nash equilibrium [27]. By showing that our user allocation game is a potential game, we can confirm the existence of a Nash equilibrium. An ordinal potential game [27] can be defined as follows.

Definition 4. ***(Potential Game)*** *A game is an ordinal potential game if, for a potential function $\phi(\boldsymbol{a})$, there exists*

$$C_{\boldsymbol{a}_{-i},\boldsymbol{p}}(\boldsymbol{a}_i) > C_{\boldsymbol{a}_{-i},\boldsymbol{p}}(\boldsymbol{a}_i') \Leftrightarrow \phi_{\boldsymbol{a}_{-i}}(\boldsymbol{a}_i) > \phi_{\boldsymbol{a}_{-i}}(\boldsymbol{a}_i') \tag{18}$$

where $u_i \in \mathcal{U}$, $\boldsymbol{a}_i, \boldsymbol{a}'_i \in \mathcal{A}_i$ *and* $\boldsymbol{a}_{-i} \in \prod_{q \neq i} \mathcal{A}_q$.

The following theorem proves that our user allocation game is an ordinal potential game.

Theorem 1. *The formulated user allocation game* Γ *is an ordinal potential game with the potential function:*

$$\begin{aligned}\phi(\boldsymbol{a}) = & \sum_{u_i \in U} \eta_1 \sum_{t \in \mathcal{T}} \left(\tau^t w_i^t \log_{z^t}(|\mathcal{U}_j|)\right) I_{\{\sum_{c_j^k \in \mathcal{C}_j} \boldsymbol{a}_{j,i}^k = 1\}} \\ & + \sum_{u_i \in U} \eta_2 h_{j,i}^k \sum_{q=i+1}^{|\mathcal{U}_j^k|} p_{j,i}^k I_{\{\boldsymbol{a}_q = \boldsymbol{a}_i\}} I_{\{\sum_{c_j^k \in \mathcal{C}_j} \boldsymbol{a}_{j,i}^k = 1\}} \\ & + \sum_{u_i \in U} \eta_2 \sum_{s_l \in \mathcal{S} \setminus \{s_j\}} (|h_{l,i}^k|^2 p_l^k) I_{\{\sum_{c_j^k \in \mathcal{C}_j} \boldsymbol{a}_{j,i}^k = 1\}} \\ & + \sum_{u_i \in U} \left(\epsilon \sum_{t \in \mathcal{T}} (\tau^t w_{max}^t) + \varepsilon I_{max}\right) I_{\{\sum_{s_j \in \mathcal{S}} \sum_{c_j^k \in \mathcal{C}_j} \boldsymbol{a}_{j,i}^k = 0\}}\end{aligned} \tag{19}$$

where $I_{\{condition\}}$ *is an indicator function that returns 1 if the condition is true, and 0 otherwise.*

Proof: Please refer to Appendix **??**. □

5.2 Decentralized User Allocation Algorithm

Given the user allocation game formulated above, we propose a **m**ulti-tenancy and **i**nterference-aware **u**ser **a**llocation algorithm (miUA) for finding a Nash equilibrium in the game. It is an iterative and decentralized algorithm that follows a class of strategy updating rules called *best response dynamics* [28], which is an evolutionary process involving a finite number of iterations. In each iteration, the decision for each user is determined by its best responses (the allocation decisions that incur the lowest system costs) to the decisions for other users made in the previous iteration. This is a decentralized process where the decision-making process for each user can be executed in parallel. Due to the *Finite Improvement Property* of ordinal potential games, it is guaranteed that this process will eventually converge to a Nash equilibrium [27].

Given a set of edge servers $\mathcal{S}$, users $\mathcal{U}$, and other required parameters, Algorithm 1 allocates users so that the total system cost (14) is minimized. During the allocation, every user is assumed to have a fixed and identical transmit power p initially. Once all the users have been allocated, each user's transmit power will be adjusted to meet the data rate requirement. This will be discussed in Section 6. With regards to the user allocation, no user is allocated initially. In other words, their initial decisions are $\mathbf{a}_i = (s_j, c_j^k) = (0, 0)$, $\forall u_i \in \mathcal{U}$ (Lines 1-3). After that, decisions are updated for users iteratively (Lines 5-16) such that the system cost incurred in the next iteration is lower than the previous iteration. The updated decision for user u_i's is denoted as $\mathbf{a}'_i$, which incurs a new system cost $C_{\mathbf{a}',\mathbf{p}}$. Once all the decision updates are submitted, the decision that incurs the lowest system cost (Line 17) will be chosen and applied to the corresponding user (Line 18). The allocation strategy $\mathbf{a}$ will also be updated accordingly. Note that the allocation strategy $\mathbf{a}$ at this stage is not final and can be updated in the consequent iterations. After this iteration, users that are not affected by the updated allocation strategy are not required to alter their current decisions. The decisions for users affected by the latest allocation strategy update need to be updated. For example, say users u_1 and u_2 both want to be allocated to the same channel on the same edge server. After user u_2 is allocated to this server in the current iteration of the game, this server is now exhausted of computing resources and insufficient to server user u_1. Consequently, the decision for user u_1 needs to be updated with a new pair of edge server and channel.

Algorithm 1 miUA

Input: $\mathcal{S}$, $\mathcal{U}$, and other parameters
Output: user allocation strategy $\mathbf{a}$

1: **initialization:**
2: every user u_i is initially unallocated, $\mathbf{a}_i = (s_j, c_j^k) = (0, 0)$, $\forall u_i \in \mathcal{U}$.
3: **end initialization**
4: **repeat**
5: Get the current system cost $C_{\mathbf{a},\mathbf{p}}$
6: **for** each user $u_i \in \mathcal{U}$ **do**
7: **for** each user u_i's neighbor server $s_j \in \mathcal{S}_i$ **do**
8: **for** each server s_j's channel $c_j^k \in \mathcal{C}_j$ **do**
9: Calculate $C_{\mathbf{a}',\mathbf{p}}(\mathbf{a}'_i)$ – the new cost if user u_i is allocated to channel c_j^k in server s_j
10: **end for**
11: **end for**
12: From all possible decisions $\mathbf{a}'_i$ above, find one with the lowest cost $C_{\mathbf{a}',\mathbf{p}}(\mathbf{a}'_i)$
13: **if** $\mathbf{a}'_i \neq \mathbf{a}_i$ and $C_{\mathbf{a}',\mathbf{p}} < C_{\mathbf{a},\mathbf{p}}$ **then**
14: Contend $\mathbf{a}'_i$ for the decision update
15: **end if**
16: **end for**
17: Find user u_i, whose decision update $\mathbf{a}'_i$ incurs system cost $C_{\mathbf{a}',\mathbf{p}}$ that is the lowest among all users $\mathcal{U}$
18: Apply decision $\mathbf{a}'_i$
19: **until** no decision updates needed for any users
20: Execute Algorithm 2 to allocate transmit power to users

The process for finding the best allocation decision for each user (Lines 5-16) will now be discussed in more detail. In each iteration, the best allocation decision for each user u_i is determined by going through every channel on every u_i's neighbor edge server (Lines 7-11). The pair of channel and edge server that incurs the lowest system cost, which can be calculated by (14) (Line 9), is selected for user u_i (Line 12). Next, if user u_i is not already allocated to this channel on this edge server, and the new system cost is lower than the current system cost, this pair of channel and edge server will be submitted for the decision update opportunity (Lines 13-15). If selected (by the mechanism previously discussed, Line 17), the decision update for user u_i will be applied. Should a better allocation decision for user u_i be found in consequent iterations, it will again be submitted to be updated. It is important to note that the decision update process for each user (Lines 6-11) can be executed in parallel because the processes for different users are independent of each other. Moreover, for each individual user, the search for the best pair of channel and edge server (Lines 7-11) can also be parallelized.

After the user allocation process has been completed, Algorithm 2 will be executed to allocate transmit power to

users (Line 20) to ensure their required data rate.

5.3 Performance Analysis

5.3.1 Convergence Analysis

The convergence time of our game is measured by the number of iterations T taken by the game to reach a Nash equilibrium. Theorem 2 proves the upper bound of the convergence time.

Theorem 2. *The convergence time of Algorithm 1 is upper bounded by:*

$$T \leq \frac{N\left(\epsilon \sum_{t\in\mathcal{T}}(\tau^t w_{max}^t) + \varepsilon I_{max}\right)}{|h_{min}|^2 p} \tag{20}$$

where $|h_{min}|^2$ is the minimum channel gain that a user could experience, and p is the default transmit power allocated to users during the user allocation process.

Proof: Please refer to Appendix **??**. □

5.3.2 Performance Bounds

Theorem 3 analyzes the lower bound and the upper bound on the number of allocated users on each edge server.

Theorem 3. *Let $num_j(\boldsymbol{a})$ denote the number of users allocated to edge server s_j given a user allocation strategy $\boldsymbol{a}$, we have:*

$$\left\lfloor \frac{Q_j^t}{w_{max}^t} \right\rfloor - 1 \leq num_j(\boldsymbol{a}) + \frac{\log_{z^t}(num_j(\boldsymbol{a})!)}{100} \leq \left\lfloor \frac{Q_j^t}{w_{min}^t} \right\rfloor \tag{21}$$

Proof: Please refer to Appendix **??**. □

Given the bounds on the number of allocated users, one can then derive the bounds on the system cost by applying the bounds to $C_{\mathbf{a},\mathbf{p}}$ (Eqs. (14), (12), and (15)).

6 Power Allocation

After the user allocation is finished, miUA allocates transmit power to users to ensure their required data rate. The power allocation problem formulation can be found in Section 4.2.

6.1 Power Allocation Problem Transformation

Instead of allocating transmit power to each user individually, the power allocation problem (16) can be converted into a problem of finding the total transmit power of all the users on a channel (or in other words, the transmit power allocated to a channel). After that, the power allocated to each channel will be allocated to the users on that channel. The rationale behind this is that the transmit power required by a user is determined by the total transmit power of users allocated to its neighbor edge servers on the same channel, apart from the transmit power of users sharing the channel on the same server. Once the total transmit power of the users allocated to the neighbor edge servers on the same channel is found, we can find the transmit power for each individual users. Lemma 2 below defines the minimum transmit power required by a channel to satisfy the data rate requirement of the users allocated to that channel. We use $\mathbf{p}_{sc} = \{p_j^k | c_j^k \in \mathcal{C}_j, s_j \in \mathcal{S}\}$ to denote the channel transmit power allocation strategy.

Lemma 2. *Given $\mathcal{U}_j^k$ – the set of users allocated to channel c_j^k on edge server s_j, to ensure the required data rate Υ for all those users, the total transmit power p_j^k of those users, i.e., the transmit power allocated to channel c_j^k, must satisfy:*

$$p_j^k \geq \sum_{i=1}^{|\mathcal{U}_j^k|} \left(2^{\frac{\Upsilon}{B_j^k}} - 1\right)\left(2^{\frac{\Upsilon}{B_j^k}} - 1\right)^{i-1} H_{j,i}^k \triangleq y_j^k(\boldsymbol{p}_{sc})$$

Proof: Please refer to Appendix **??**. □

According to [25], [29], the optimal solution to finding the total transmit power of all the users allocated to a channel can be found by solving the following problem, which is transformed from problem (16).

$$\min_{\{\mathbf{p}_j^k\}} \sum_{j=1}^{M}\sum_{k=1}^{V} p_j^k \tag{22a}$$

$$\text{s.t. } p_j^k \geq \sum_{i=1}^{|\mathcal{U}_j^k|} \left(2^{\frac{\Upsilon}{B_j^k}} - 1\right)\left(2^{\frac{\Upsilon}{B_j^k}} - 1\right)^{i-1} H_{j,i}^k \triangleq y_j^k(\mathbf{p}_{sc}),$$
$$\forall c_j^k \in \mathcal{C}_j, \forall s_j \in \mathcal{S} \tag{22b}$$

$$\Theta(c_j^k), \forall s_j \in \mathcal{S} \tag{22c}$$

$$\sum_{k=1}^{V} p_j^k \leq P_j, \forall s_j \in \mathcal{S} \tag{22d}$$

$$p_j^k \in \mathbb{R}_{\geq 0}, \forall c_j^k \in \mathcal{C}_j, \forall s_j \in \mathcal{S} \tag{22e}$$

where objective (22a) minimizes the total transmit power allocated to all the channels on all the edge servers. Constraint (22b) is retrieved from Lemma 2, which helps enforce constraint (13e). Constraint (22c) enforces the SIC decoding order on each channel. Constraint (22d) ensures that the total transmit power allocated to all the channels on an edge server does not exceed that edge server's power capacity. Constraint (22e) indicates the possible values of channel transmit power decision p_j^k.

6.2 Decentralized Power Allocation Algorithm

Similar to Algorithm 1, the power allocation algorithm introduced below (Algorithm 2) is also decentralized.

This is a two-stage algorithm. First, the BS's transmit power is allocated to each channel (Lines 1-11). After that, the transmit power allocated to each channel will be allocated to the users on that channel (Lines 12-19). In the first stage, initially, the transmit power of a BS is equally assigned to all channels on that BS (Lines 2-4). This is followed by an iterative and recursive process for updating the transmit power allocated to channels, which consists of a finite number of iterations (Lines 5-11). In each iteration, the transmit power allocated to each channel is updated based on the transmit power allocated to other channels, which may have been updated in the previous iteration (Line 8). Similar to [25], $y_j^k(\mathbf{p}_{sc})$ is a *standard interference function* [29] since it satisfies three criteria as follows: 1) Positivity: $y_j^k(\mathbf{p}_{sc}) > 0$, and 2) Monotocity: If $\mathbf{p}_{sc} > \mathbf{p}'_{sc}$ then $y_j^k(\mathbf{p}_{sc}) > y_j^k(\mathbf{p}'_{sc})$, and 3) Scalability: For all $\alpha > 1$, then $\alpha y_j^k(\mathbf{p}_{sc}) > y_j^k(\alpha\mathbf{p}_{sc})$. When $y_j^k(\mathbf{p}_{sc})$ is standard, Algorithm 2 is referred to as a *standard power control algorithm*, which will eventually converge to a unique fixed point (the global optimal solution, if one exists) from any initial power allocation [25], [29], [30].

In the second stage, the transmit power allocated to channels will be allocated to the users on these channels

Algorithm 2 Decentralized power allocation algorithm

Input: $\mathcal{S}$, $\mathcal{U}$, $\mathbf{a}$, and other parameters
Output: power allocation strategy $\mathbf{p}$

1: **Stage 1:** allocating BS's transmit power to channels
2: **initialization:**
3: $p_j^k = P_j/V, \forall s_j \in \mathcal{S}, \forall c_j^k \in \mathcal{C}_j$
4: **end initialization**
5: **repeat**
6: **for** each server $s_j \in \mathcal{S}$ **do**
7: **for** each server s_j's channel $c_j^k \in \mathcal{C}_j$ **do**
8: Calculate $p_j^{k^{(iIteration)}} = y_j^k(\mathbf{p}_{sc}^{(iIteration-1)})$
9: **end for**
10: **end for**
11: **until** convergence
12: **Stage 2:** allocating channel's transmit power to users
13: **for** each server $s_j \in \mathcal{S}$ **do**
14: **for** each server s_j's channel $c_j^k \in \mathcal{C}_j$ **do**
15: **for** $u_i \in \mathcal{U}_j^k$ **do** ▷ start from the weakest user
16: Calculate $\mathbf{p}_{j,i}^k = \left(2^{\frac{\Upsilon}{B_j^k}} - 1\right)\left(\sum_{q=1}^{i-1} \mathbf{p}_{j,q}^k + H_{j,i}^k\right)$
17: **end for**
18: **end for**
19: **end for**

(Lines 13-19). On each channel, the transmit power is allocated in the order of channel conditions, or the SIC decoding order (8). The user with the weakest channel condition is the first to be allocated transmit power using (9), where $r_{j,i}^k = \Upsilon$ (Line 16). Transmit power is allocated to that user first because it is the first to decode the received signal without the need for SIC or the consideration of the power of the other users sharing the same channel.

7 Performance Evaluation

We perform a series of experiments to evaluate the performance of miUA against state-of-the-art and baseline approaches.

7.1 Experimental Settings

In the experiments, we employ a 19-hexagonal macrocell model, i.e., $M = 19$. The experimental settings are compliant with the existing LTE specifications [31] and summarized in Table 1. Edge servers' available computing resources Q_j are randomly generated by following a normal distribution $\mathcal{N}(\mu, 10^2)$, where μ is the average capacity of each resource type in $\mathcal{T} = \{\text{CPU}, \text{GPU}, \text{RAM}, \text{storage}\}$, and the standard deviation is 5. We set the weight parameters $\eta_1 = \eta_2 = 0.5$. Users are randomly located within the coverage of those edge servers by following a uniform distribution. Users' required data rate Υ is set at 2Mbps. We assume that users have three possible levels of normalized computing resource requirements, $w_i \in \{< 1, 2, 1, 1 >, < 2, 1, 2, 2 >, < 3, 3, 2, 2 >\}$. We have conducted experiments with other resource requirements and achieved similar results. Thus, we select those three levels as the representatives. Each user's computing resource requirement is randomly selected from those three levels.

We conduct two sets of experiments. In experiment Set #1, the average computing resource capacity μ of edge servers is fixed at 16; and we vary the number of users N from 200 users to 600 users in steps of 50. In experiment Set #2, the number of users is fixed at 500; and we vary the average computing resource capacity μ of edge servers from 10 to 26 in steps of 2. To evaluate the performance of miUA in achieving the optimization objective, i.e., minimizing the system cost (computing resource cost), we compare the normalized computing resource costs achieved by the five approaches. In addition, we compare the number of users allocated to edge servers, the higher the better. The convergence time of miUA is also evaluated, which is a critical machine-independent efficiency indicator for game-theoretical approaches [5], [19], [20], [32], [33].

TABLE 1: Experimental Settings

Cell layout	Hexagonal grid, 19 cell sites (edge servers)
Cell radius (R_j)	289m
Inter-site distance	500m
Minimum distance between user and edge server	35m
Large-scale path loss model	$128.1 + 37.6log_{10}(d_{j,i})$dB
BS maximum transmit power (P_j)	46dBm
Thermal noise density	-174dBm/Hz
System bandwidth (B)	10MHz
Number of channels (V)	5

7.2 Performance Benchmark

We compare miUA against four representative approaches, including two state-of-the-art approaches and two baseline approaches:

- ***SCG-SA*** [34]: This approach solves the user allocation problem in a NOMA-based cellular network with the objective to improve the system energy efficiency. SCG-SA ranks users based on their channel conditions. Users with strong channel conditions are allocated first because they consume less transmit power. However, SCG-SA is designed to operate in a pure cellular network without edge servers, and thus does not consider the heterogeneity of edge servers with varying computing resources.
- ***EUAGame*** [5]: This approach solves the user allocation problem in an MEC system with the objective to maximize the number of allocated users at minimum computing resource costs by leveraging the multi-tenancy feature. However, EUAGame is not designed to operate in a multi-channel cellular system or a NOMA-based system. It completely neglects the communication/networking aspect. In the experiments, after users are allocated to edge servers, this approach first allocates users to channels randomly and then performs a fixed power allocation as adopted in [11], [35].
- ***NearestUA***: We propose a naive baseline approach that allocates each user to their nearest edge server with sufficient computing resources. The rationale behind this approach is that a short distance between a user and an edge server usually results in a strong

channel condition. After that, each user is allocated to the channel with the fewest users. Then, this approach employs Algorithm 2 to allocate transmit power to users.

- ***Random***: This baseline approach allocates each user to a random edge server and channel as long as that edge server has sufficient computing resources. Similar to NearestUA, this approach also employs Algorithm 2 to allocate transmit power to users.

7.3 Experimental Results

Figures 2, 3, and 4 demonstrate the results of experiment Set #1. Figures 5, 6, and 7 demonstrate the results of experiment Set #2. In general, miUA significantly outperforms all other approaches in both sets of experiments, being able to allocate the most users at the lowest system cost. SCG-SA follows behind miUA by a large margin, but outperforms EUAGame, NearestUA, and Random significantly. Fig.s 4 and 7 demonstrate miUA's convergence time and the impacts of the number of users as well as the available server capacity.

7.3.1 Effectiveness

Experiment Set #1: In this experiment set, the number of users gradually increases from 200 to 600. As the number of users increases, the percentage of users allocated by all the approaches decreases (Fig. 2). As the amount of computing resources available is fixed, adding more users to the system will exhaust edge servers' computing resources quickly, thus increasing the number of users that cannot be allocated to any edge servers. **Under all parameter settings, miUA is able to allocate the most users**. SCG-SA comes second with the percentage of users allocated being 10%-20% lower than miUA. EUAGame, NearestUA, and Random produce roughly similar results with the percentage of users allocated being far lower than miUA (20%-35% lower). From Fig. 3 we can also see that **the system cost incurred by miUA is the lowest among all the approaches**. As the number of users increases, the system cost difference between miUA and other approaches increases, indicating miUA's more efficient use of computing resources than other approaches. Although SCG-SA significantly allocates more users than EUAGame, NearestUA, and Random, it is only marginally better than those three approaches in terms of the system cost. This shows that SCG-SA is not capable of utilizing the edge servers' computing resources by leveraging multi-tenancy.

Experiment Set #2: In this set of experiments, we vary the average computing resource capacity μ available on edge servers from 10 to 26. As the computing resources become more abundant, more users can be allocated to edge servers, leading to an increasing trend in the percentage of users allocated by all the approaches (Fig. 5). Again, it can be seen that **miUA vastly outperforms other approaches**. It is able to allocate approximately 10%-20% more users than SCG-SA, and 15%-40% more users than EUAGame, NearestUA, and Random. EUAGame, NearestUA, and Random have roughly similar performance, where EUAGame is just slightly better than NearestUA, which faintly outperforms Random. In Fig. 6, it can be seen that **miUA and EUAGame incur considerably lower system costs than the other approaches**.

Discussion: In the previous discussion (Fig.s 2 and 5), we refer to the allocated users as the users who are allocated to edge servers with sufficient computing resources and data rate. In this section, we analyze the allocated users at a higher granularity level (Fig.s 8 and 9). The users are categorized into four groups: 1) unallocated users, including users that are not allocated to any edge servers and users that are allocated to edge servers but receiving insufficient computing resources and data rate, 2) users that are allocated to edge servers, receiving required data rate but insufficient computing resources, 3) users that are allocated to edge servers, receiving sufficient computing resources but insufficient data rate, and 4) users that have all the requirements satisfied to use the services provided in edge servers. We compare the two state-of-the-art approaches, SCG-SA and EUAGame, to show the importance of an effective power control scheme or computing resource allocation. SCG-SA is designed to allocate users in a NOMA-based cellular network but does not consider the scarcity and heterogeneity of computing resources in MEC systems. In contrast, EUAGame is designed to allocate users in an MEC system but ignores its communication aspect, including multiple wireless channels, interference, and power control. As expected, we can observe in Set #1 (Fig. 8) that there is a large number of users experiencing insufficient computing resources when allocated by SCG-SA. With EUAGame, there is a large number of users receiving insufficient data rates, because EUAGame does not consider interference when allocating users and lacks a proper power control, which is critical in a NOMA system. We can observe the same in Set #2 (Fig. 9).

In general, **SCG-SA outperforming EUAGame indicates the significance of considering wireless interference and proper power control in a NOMA-based MEC system**. Our proposed approach miUA considers both aspects of a NOMA-based MEC system, i.e., the communication aspect (multiple wireless channels with different types of interference, and power control), and the computation aspect (the scarcity and heterogeneity of computing resources on edge servers). Therefore, **miUA clearly outperforms all other approaches**. NearestUA and Random, although leveraging a state-of-the-art power control mechanism, are still remarkably outperformed by all other approaches. This shows that not systematically incorporating wireless interference and computing resource cost leads to a poor performance in allocating users in a NOMA-based MEC system.

7.3.2 Efficiency

In order to evaluate the efficiency of miUA, we measure its convergence time by the number of decision iterations it takes to reach a Nash equilibrium (Fig.s 4 and 7). In Set #1, the number of iterations slowly increases with the increase in the number of users to be allocated (Fig. 4). Since one user is allocated in each decision iteration, we analyze the ratio between the number of iterations and the number of allocated users (the red line in the graph). We can see that the ratio starts from 1.5 at 200 users, then goes down to around 1.2 at 600 users. This indicates that when the number of users is low, users' decisions tend to change more often as there is more room for improvement. When the number

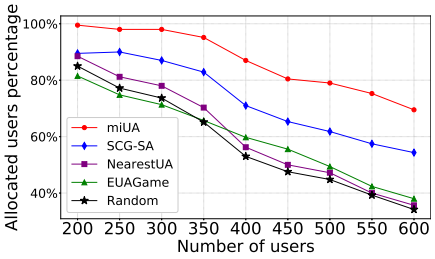


Fig. 2: Percentage of allocated users vs. number of users (Set #1).

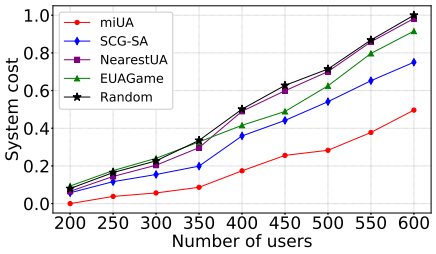


Fig. 3: System cost vs. number of users (Set #1).

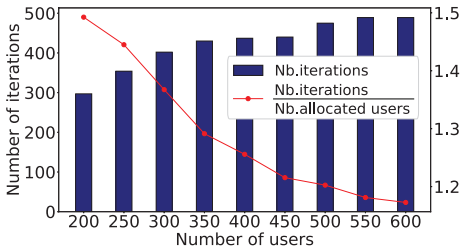


Fig. 4: Number of decision iterations vs. number of users (Set #1, miUA).

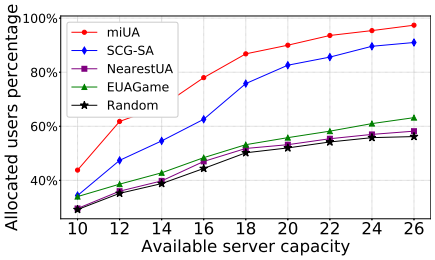


Fig. 5: Percentage of allocated users vs. number of users (Set #2).

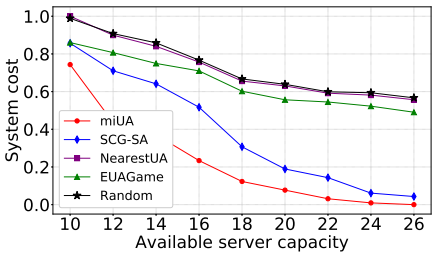


Fig. 6: System cost vs. number of users (Set #2).

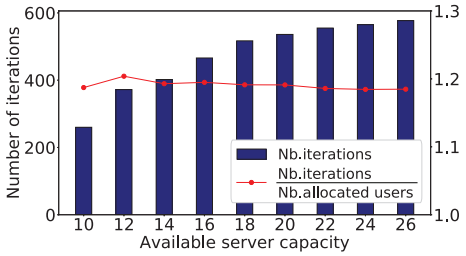


Fig. 7: Number of decision iterations vs. number of users (Set #2, miUA).

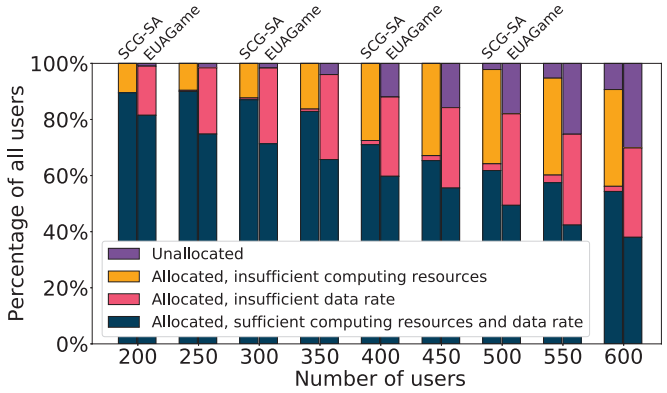


Fig. 8: Details of allocated users (Set #1).

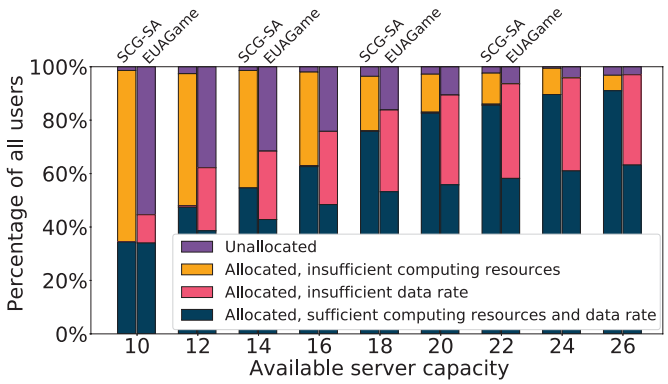


Fig. 9: Details of allocated users (Set #2).

of users is large, edge servers and channels tend to be more occupied, hence less room for decision updates.

In Set #2, where the number of users is fixed, we can see that miUA's convergence time increases with the increase in the amount of computing resources available (Fig. 7). As we increase the amount of available computing resources, there is a high probability that users are allocated to edge servers with sufficient resources. The number-of-iterations-to-number-of-allocated-users ratio remains relatively stable at around 1.2. This indicates that despite the amount of the available computing resources, users do not change their decisions very often. We can conclude that the convergence time is more dependent on the number of users (Set #1, Fig. 4).

8 Related Work

Mobile edge computing (MEC) is a new computing paradigm that brings together cloud computing and cellular network. Deploying edge servers at cellular base stations allows end-users to use the services and applications provided by app vendors with low network latency.

8.1 User Allocation in MEC

The user allocation problem in MEC systems was first introduced in [15], [16], in which the authors propose an optimal approach and an efficient heuristic to allocate as many users to as few edge servers as possible. In [36], [37], the authors attempt to solve the user allocation problem in which a user's computing resource requirement can be dynamically adjusted during the allocation process. Their objective is to maximize the users' quality of experience. The authors of [14] tackle the scenario where users can move between different edge servers, which requires re-allocating users from one server to another. They aim to minimize the number of reallocations. The authors of [5] aim to minimize the system cost measured by the amount of computing resources consumed by users, which is similar to our objective. The authors of [38] propose a user allocation approach that takes into account the wireless interference, aiming to improve users' data rates. We, on the

other hand, try to minimize the computing resource cost. The data rate in our work is satisfactory once it reaches a certain level. Among the aforementioned studies, only the study presented in [38] incorporates the communication aspect of an MEC system that features multiple wireless channels with interference. Nevertheless, it only considers intra-cell interference, whereas we also consider inter-cell interference, which could be very severe in a dense cellular network. Furthermore, none of the existing work on EUA considers NOMA, in which interference and power control play an important role and should not be overlooked as demonstrated in our experiments.

Realizing the benefit provided by NOMA, researchers are beginning to study MEC problems under NOMA settings [9], [39], [40]. However, most of them focus on the computing offloading problem, which is an important problem that shares some similarities with the edge user allocation problem. Nevertheless, they are distinguished by several essential characteristics. In the computation offloading problem, a user generates a series of computation tasks, which can be executed partly on its local device or edge servers (partial offloading), or completely on edge servers (full offloading) [1]. A computation task usually has a single-dimensional resource requirement (CPU cycles) [17], [21], [40], [41]. While in the edge user allocation problem, an app vendor needs to hire multiple types of computing resources to serve a user on an edge server [5], [42], [43], and the resources allocated to a user must be available at all times during the user's connection to the edge server. In addition, in some studies [21], [44], users are assumed to be pre-allocated to edge servers before their computation tasks are offloaded. Furthermore, computation offloading is a low-level problem that is often tackled from the edge infrastructure provider's (or the mobile network operator's) perspective, whose main objective is to minimize either the overall system delay or the system energy consumption [1], [40]. On the other hand, the edge user allocation problem is tackled from the app vendor's perspective, who aims to maximize their profit by serving as many users as possible at the lowest costs.

8.2 User Allocation in NOMA-based Cellular Network

The user allocation problem (also often referred to as the user association problem) is a mature and well-researched problem in conventional cellular networks [45]. However, the introduction of NOMA has sparked a new wave of research on this problem. The authors of [10] aim to maximize the sum-throughput in both downlink and uplink NOMA systems. They also demonstrate that NOMA can achieve a significant throughput gain over the traditional orthogonal multiple access (OMA) scheme. Inter-cell interference is not yet considered in [10] though. The authors of [11] confirm that carefully pairing users into channels in NOMA can offer a larger sum rate than OMA. Nevertheless, this work is limited to only two users per channel. The authors of [12] try to maximize the sum data rate in a multi-cell but single-channel system. The authors of [13] aim to minimize the power consumption in a multi-channel single-cell NOMA scenario. If applied in a multi-cell NOMA scenario, the authors would also have to consider the inter-cell interference. The authors of [34] attempt to increase the energy efficiency in a multi-cell multi-channel NOMA system by allocating users based on a ranking of channel conditions. In [35], the authors try to improve users' quality of experience when allocating users. From the review of existing studies, we can see that all the existing approaches will need to be remodeled or redefined when adapting to a new environment like MEC as they completely neglect the computing side of MEC (the scarcity and heterogeneity of computing resources on edge servers). Our proposed approach eliminates this limitation by jointly considering both communication and computation aspects in an MEC environment.

9 Conclusion and Future Work

In this paper, the edge user allocation problem in a downlink non-orthogonal multiple access (NOMA) based mobile edge computing (MEC) system is investigated from the app vendor's perspective. To attack this $\mathcal{NP}$-hard problem, we formulate the problem as a potential game with the objective to maximize the number of allocated users at a minimum computing resource cost. By jointly considering both the communication and computation aspects of a NOMA-based MEC system, miUA, our decentralized game-theoretic approach, greatly outperforms the state-of-the-art and baseline approaches, being able to serve the most users with sufficient data rate and computing resources. Our experiments highlight the significance of incorporating both wireless interference and computing resource consumption into the user allocation approach in a NOMA-based MEC system. We also theoretically analyze the optimality and convergence of our proposed approach.

Integrating and utilizing NOMA in MEC is still in a very early research stage, particularly for the edge user allocation problem. There is plenty of future work that could be studied, for instance, how to handle handover when dealing with user mobility, dynamic user arrivals and departures, incorporating a more specific pricing model imposed on app vendors, etc. Furthermore, solutions to the user problem in uplink NOMA-based MEC systems should also be investigated.

Acknowledgments

This research is partly funded by Australian Research Council Discovery Project grants (DP170101932 and DP180100212), and Laureate Fellowship FL190100035.

References

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[2] "Ericsson Mobility Report," *Ericsson, Stockholm*, 2019. [Online]. Available: www.ericsson.com/4acd7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf

[3] A. Benjebbour, Y. Saito, Y. Kishiyama, A. Li, A. Harada, and T. Nakamura, "Concept and practical considerations of non-orthogonal multiple access (NOMA) for future radio access," in *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems*. IEEE, 2013, pp. 770–774.

[4] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proceedings of IEEE Vehicular Technology Conference*. IEEE, 2013, pp. 1–5.

[5] Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, and Y. Yang, "A game-theoretical approach for user allocation in

edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 515–529, 2020.

[6] F. Chong and G. Carraro, "Architecture strategies for catching the long tail," *MSDN Library, Microsoft Corporation*, pp. 9–10, 2006.

[7] B. P. Rimal and M. Maier, "Workflow scheduling in multi-tenant cloud computing environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 290–304, 2016.

[8] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proceedings of HotPower Workshop on Power Aware Computing and Systems*. USENIX, 2008, pp. 10–15.

[9] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 375–390, 2018.

[10] M. S. Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems," *IEEE Access*, vol. 4, pp. 6325–6343, 2016.

[11] Z. Ding, P. Fan, and H. V. Poor, "Impact of user pairing on 5G nonorthogonal multiple-access downlink transmissions," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6010–6023, 2015.

[12] K. Wang, Y. Liu, Z. Ding, A. Nallanathan, and M. Peng, "User association and power allocation for multi-cell non-orthogonal multiple access networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5284–5298, 2019.

[13] F. Guo, H. Lu, D. Zhu, and H. Wu, "Interference-aware user grouping strategy in NOMA systems with QoS constraints," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2019, pp. 1378–1386.

[14] Q. Peng, Y. Xia, Z. Feng, J. Lee, C. Wu, X. Luo, W. Zheng, H. Liu, Y. Qin, and P. Chen, "Mobility-aware and migration-enabled online edge user allocation in mobile edge computing," in *Proceedings of IEEE International Conference on Web Services*. IEEE, 2019, pp. 91–98.

[15] P. Lai, Q. He, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *Proceedings of International Conference on Service-Oriented Computing*. Springer, 2018, pp. 230–245.

[16] P. Lai, Q. He, J. Grundy, F. Chen, M. Abdelrazek, J. Hosking, J. Grundy, and Y. Yang, "Cost-effective app user allocation in an edge computing environment," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020, doi: 10.1109/TCC.2020.3001570.

[17] Z. Hong, W. Chen, H. Huang, S. Guo, and Z. Zheng, "Multi-hop cooperative computation offloading for industrial IoT–edge–cloud computing environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2759–2774, 2019.

[18] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, 2018.

[19] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.

[20] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, no. 5, pp. 2795–2808, 2016.

[21] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1619–1632, 2018.

[22] S. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 721–742, 2016.

[23] J. Choi, "Minimum power multicast beamforming with superposition coding for multiresolution broadcast and application to NOMA systems," *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 791–800, 2015.

[24] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, "On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users," *IEEE Signal Processing Letters*, vol. 21, no. 12, pp. 1501–1505, 2014.

[25] Z. Yang, C. Pan, W. Xu, Y. Pan, M. Chen, and M. Elkashlan, "Power control for multi-cell networks with non-orthogonal multiple access," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 927–942, 2017.

[26] M. R. Garey and D. S. Johnson, *Computers and intractability*. Freeman San Francisco, 1979, vol. 174.

[27] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[28] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT Press, 1994.

[29] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1341–1347, 1995.

[30] Y. Fu, Y. Chen, and C. W. Sung, "Distributed power control for the downlink of multi-cell noma systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6207–6220, 2017.

[31] E. 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) requirements for LTE Pico Node B (3GPP TR 36.931 version 9.0.0 Release 9)," Tech. Rep., 2011.

[32] B. Yang, Z. Li, S. Chen, T. Wang, and K. Li, "Stackelberg game approach for energy-aware resource allocation in data centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3646–3658, 2016.

[33] S. Ma, S. Guo, K. Wang, W. Jia, and M. Guo, "A cyclic game for joint cooperation and competition of edge resource allocation," in *Proceedings of International Conference on Distributed Computing Systems*. IEEE, 2019, pp. 503–513.

[34] H. Zeng, X. Zhu, Y. Jiang, Z. Wei, and T. Wang, "A green coordinated multi-cell noma system with fuzzy logic based multi-criterion user mode selection and resource allocation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 480–495, 2019.

[35] J. Cui, Y. Liu, Z. Ding, P. Fan, and A. Nallanathan, "QoE-based resource allocation for multi-cell NOMA networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6160–6176, 2018.

[36] P. Lai, Q. He, G. Cui, X. Xia, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Edge user allocation with dynamic quality of service," in *Proceedings of International Conference on Service-Oriented Computing*. Springer, 2019, pp. 86–101.

[37] ——, "QoE-aware user allocation in edge computing systems with dynamic QoS," *Future Generation Computer Systems*, vol. 112, pp. 684–694, 2020.

[38] G. Cui, Q. He, X. Xia, P. Lai, F. Chen, T. Gu, and Y. Yang, "Interference-aware SaaS User Allocation Game for Edge Computing," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020, doi: 10.1109/TCC.2020.3008448.

[39] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116 974–117 017, 2020.

[40] F. Wang, J. Xu, and Z. Ding, "Optimized multiuser computation offloading with multi-antenna NOMA," in *Proceedings of IEEE Globecom Workshops*. IEEE, 2017, pp. 1–7.

[41] D. Zhang, L. Tan, J. Ren, M. K. Awad, S. Zhang, Y. Zhang, and P.-J. Wan, "Near-optimal and truthful online auction for computation offloading in green edge-computing systems," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 880–893, 2019.

[42] M. Hu, L. Zhuang, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Learning driven computation offloading for asymmetrically informed edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 8, pp. 1802–1815, 2019.

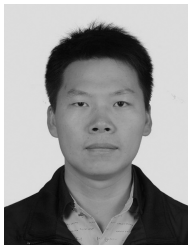
[43] Y. Liang, G. Jidong, S. Zhang, J. Wu, Z. Tang, and B. Luo, "A utility-based optimization framework for edge service entity caching," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 11, pp. 2384–2395, 2019.

[44] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2018, pp. 207–215.

[45] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 524–540, 2011.

Phu Lai received his MSc degree in Information Technology in 2017 and is currently working toward a PhD degree at Swinburne University of Technology, Australia. His research interests include software engineering, cloud computing and edge computing.

Qiang He received the first PhD degree from Swinburne University of Technology (SUT), Australia, in 2009 and the second PhD degree in computer science and engineering from Huazhong University of Science and Technology (HUST), China, in 2010. He is a lecturer at Swinburne University of Technology. His research interests include software engineering, cloud computing, and services computing. More details about his research can be found at www.sites.google.com/site/heqiang/.



Guangming Cui received his master's degree from Anhui University, China, in 2018. He is working toward the PhD degree at the Swinburne University of Technology. His research interests include software engineering, edge computing, and service computing.

Feifei Chen received her PhD degree from Swinburne University of Technology, Australia, in 2015. She is a lecturer at Deakin University. Her research interests include software engineering, cloud computing and green computing.

John Grundy is the Senior Deputy Dean for the Faculty of Information Technology and a Professor of Software Engineering at Monash University. He is a Fellow of Automated Software Engineering, Fellow of Engineers Australia, Certified Professional Engineer, Engineering Executive, Member of the ACM and Senior Member of the IEEE. His current interests include large-scale systems engineering, software engineering education, etc. More details about his research can be found at www.sites.google.com/site/johncgrundy/.

Mohamed Abdelrazek is an Associate Professor of Software Engineering and IoT at Deakin University. Mohamed has more than 15 years of the software industry, research, and teaching experience. Before joining Deakin University in 2015, he worked as a senior research fellow at Swinburne University of Technology and Swinburne-NICTA software innovation lab (SSIL). More details about his research can be found at www.sites.google.com/site/mohamedalmorsy/.

John Hosking is Dean of Science at the University of Auckland and Adjunct Professor of Computer Science at the ANU. His research interests are primarily in the Software Engineering/Software Tools area and he is an active member of the Automated Software Engineering and Visual Languages research communities. John is a Fellow of the Royal Society of New Zealand and a Member of the Ako Aotearoa Academy of Tertiary Teaching Excellence.

Yun Yang received his PhD degree from the University of Queensland, Australia, in 1992. He is a full professor at Swinburne University of Technology. His research interests include software engineering, cloud and edge computing, workflow systems, and service-oriented computing.