Online User and Power Allocation in Dynamic NOMA-based Mobile Edge Computing

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

Abstract—This study tackles the online user allocation problem in mobile edge computing (MEC) systems powered by non-orthogonal multiple access. App vendors need to determine a proper wireless channel in a base station/edge server and sufficient transmit power for every user. We consider a stochastic MEC system where users arrive and depart over time. When an edge server runs out of computing resources, some users will have to wait until the resources become available again, which incurs an allocation delay cost. This cost is often not investigated in many studies, which also do not consider a multi-cell, multi-channel system as we do in this work, due to its complexity. We aim to minimize the allocation delay and transmit power costs, increasing the system's energy efficiency. To achieve this objective while guaranteeing users' data rate requirements over time, we adopt the Lyapunov framework to convert this long-term optimization problem into a series of subproblems to be solved in every time slot. To solve the aforementioned subproblems efficiently, we present a distributed game theory-based approach. The proposed algorithm is theoretically evaluated and experimentally demonstrated to outperform several baseline and state-of-the-art methods, highlighting the significance of systematic consideration for both computation and communication aspects of this problem.

Index Terms—Discrete power allocation, online user allocation, mobile edge computing, non-orthogonal multiple access (NOMA)

1 Introduction

Mobile edge computing (MEC) fuels the potential of latency-sensitive applications (smart cities, IoT, critical monitoring systems) as edge servers can be installed at cellular base stations (BSs) in close distance to users. App vendors can rent computing resources in edge servers and host their services for their users to access. To facilitate the massive connectivity over 5G/6G networks, non-orthogonal multiple access (NOMA) is proposed [1]. Compared to traditional multi-access methods for wireless communication (e.g., OFDMA, TDMA, or CDMA), NOMA achieves greater spectral efficiency and user throughput performance by accommodating multiple users concurrently with the same frequency or time resources in the power or code domain [1]. Integrating NOMA into MEC systems will further promote latency-sensitive applications in the 5G/6G era.

The *edge user allocation (EUA)* problem has been investigated extensively in recent years as an offline problem [2], [3], [4], [5], [6], [7]. MEC researchers have begun to study computation offloading with NOMA. However, the EUA problem in NOMA-based MEC still remains open. Here, we study an *online* EUA problem in downlink multi-channel multi-cell power-domain NOMA-based MEC systems. In power-domain NOMA, a frequency channel can serve multiple users simultaneously. An app vendor needs to select a suitable subchannel in a suitable BS/edge server[1] with a sufficient amount of transmit power to serve each user and satisfy its data rate requirement. This functionality is offered in MEC systems as app vendors can now access and leverage network data such as received signal, received power, throughput, neighbor cells, QoS, etc. [8], [9]. We tackle a highly stochastic time-slotted MEC system. In every time slot, there is a random number of user arrivals and departures. All future user arrivals and departures are unknown. Applications hosted on edge servers usually serve users by processing their requests then returning a large amount of data, such as videos published by content providers or graphics rendered by VR/AR applications. Thus, this study focuses on downlink transmissions.

When allocating users, app vendors have to incorporate two types of costs. Firstly, due to the heterogeneity and limitation of edge servers' computing resources [10], new users might have to wait until existing users depart the system and free up the occupied computing resources in edge servers. This incurs an *allocation delay cost*. Secondly, the *transmit power cost* must be minimized. With the above in mind, a minimum data rate requirement must be fulfilled for as many users as possible. In NOMA, the transmit powers allocated to different users are tightly coupled and must be considered in conjunction with each other.

Existing user allocation methods often do not jointly

- *P. Lai is with Cisco-La Trobe Centre for AI and IoT and Swinburne University of Technology, Australia. E-mail: p.lai@latrobe.edu.au.*
- *Q. He, and Y. Yang are with the Department of Computing Technologies, Swinburne University of Technology, Australia. E-mail: {qhe, yyang}@swin.edu.au.*
- *F. Chen and M. Abdelrazek are with the Faculty of Science, Engineering and Built Environment, Deakin University, Australia. E-mail: {feifei.chen, mohamed.abdelrazek}@deakin.edu.au.*
- *J. Hosking is with the Faculty of Science, University of Auckland, Auckland, New Zealand. E-mail: j.hosking@auckland.ac.nz.*
- *J. Grundy is with the Faculty of Information Technology, Monash University, Australia. E-mail: john.grundy@monash.edu.*

Manuscript received Dec 23, 2021; revised March, June 2022; accepted 20 July 2022.
Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Qiang He, Feifei Chen.)
Digital Object Identifier no.

[1] The terms "edge server" and "base station" (BS) will be used interchangeably.

consider computation and communication aspects of MEC. These approaches in MEC [2], [7], [11], [12], [13] neglect the critical communication aspect such as power control, the availability of multiple subchannels, and intra-cell/inter-cell interference, especially in NOMA-based networks. This is highly uneconomical since app vendors can now utilize network data. Meanwhile, user allocation methods in pure cellular networks lack the consideration of computation aspect of MEC as edge servers have different and limited amounts of computing resources. To simplify the problem, many of those do not take into account multi-channel [14], [15] or multi-cell [16], [17], or even impose a cap on the number of users on each subchannel [18], [19], [20], unnecessarily impeding the prospect of NOMA. Furthermore, many user allocation approaches have been investigated in a static scenario where temporal system dynamics is out of the equation. Without the temporal dimension, the allocation delay cost could be very high and it would impact users' quality of experience profoundly. In this study, we overcome all these limitations. We jointly make two decisions for each user: 1) user allocation, including BS/edge server and channel assignments; and 2) power allocation, so that the allocation delay and transmit power costs are minimized while not violating a number of constraints (long-term data rate requirement, resource, and proximity constraints) over all time slots. Fig. 1 provides an illustration of the EUA problem. Our key contributions include:

- We formulate an online EUA problem in stochastic multi-channel multi-cell NOMA-based MEC. We adopt Lyapunov optimization to convert this long-term problem into a series of subproblems to be solved in individual time slots. Unlike typical adoptions of Lyapunov optimization that model target systems as queuing systems, our approach aims to stabilize users' data rates over time.
- We address a number of limitations of existing approaches in power and user allocation as identified above. In summary, they do not jointly consider the communication aspect (power control, multi-channel, multi-cell, interference), the computation aspect (heterogeneity and limitedness of computing resources), and the user dynamics (random user arrivals and departures) of an MEC system. Some even limit the number of users on a subchannel.
- We show that the subproblem mentioned above is a potential game. Due to it being an NP-hard problem, it is intractable to find an optimal solution in large-scale scenarios. To find a sub-optimal solution efficiently, or a Nash equilibrium in this game, within each time slot, we introduce a two-stage decentralized game theoretical user and power allocation algorithm, utilizing the distributed nature of MEC.
- We theoretically and experimentally evaluate the proposed approach and show that it outperforms various baseline and state-of-the-art approaches significantly.

The rest of this paper is organized as follows. Key motivations are discussed in Section 2. Section 3 models the MEC system with different types of associated costs and formulates the problem. In Section 4, we present a Lyapunov optimization-based online user allocation algorithm. As part of this online algorithm, we introduce a game-theoretical approach in Section 5. Our approach is experimentally evaluated in Section 6. In Section 7, relevant literature is reviewed. Finally, Section 8 concludes the paper and suggests some future research directions.

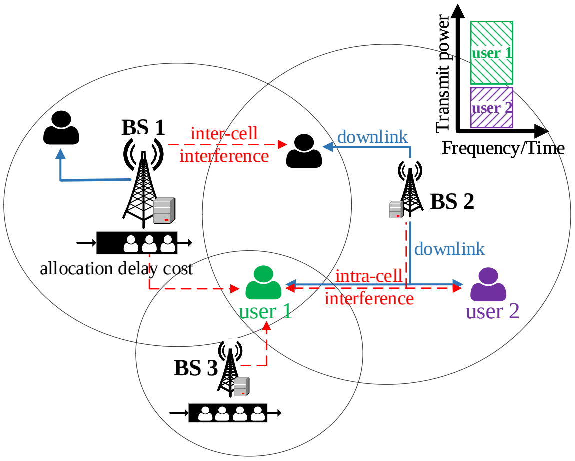

Fig. 1: An illustration of EUA problem in NOMA-based MEC

2 Motivation

A 5G cellular network is usually densely populated with many multi-channel BSs, especially in high-traffic areas (up to 50 base stations per km^2) [21], creating numerous overlapping cell coverage areas. Thus, a user may have many neighbor BSs/edge servers and experience severe inter-cell interference [15]. An app vendor needs to carefully determine which subchannel in which BS should serve the user. Compared with a usual cloud server, computing resources in an edge server are much more scarce and heterogeneous [22], [23]. Therefore, when an edge server runs out of computing resources, users allocated to this server might have to wait until some existing users depart [24]. Without incorporating this characteristic into user allocation, some of the users might have to wait for an excessive amount of time. This queuing system is very common in applications such as online gaming [25], [26], [27].

In power-domain NOMA, multiple users share the same subchannel at the same time. Still, a subchannel cannot simultaneously serve too many users because of the severe interference. App vendors thus need to find a balance in the number of users allocated to different subchannels. To decode the superposed signal transmitted from the BS, which includes the signal intended to other users on the same subchannel, each user employs a multi-user signal separation method called successive interference cancellation (SIC). In downlink transmissions, this is facilitated by varying users' transmit power. In a single-cell scenario, on a subchannel, weaker users (those with poorer channel gain) receive more transmit power than stronger users (those with better channel gain) to ensure successful decoding of the superposed signal. However, this approach is not applicable in multi-cell scenarios since a user's channel condition is partly influenced by strong inter-cell interference, which cannot be neglected.

3 System Model

We model a NOMA-based multi-channel multi-cell MEC system in this section and summarize the key notations in Appendix A of the supplementary file.

3.1 System Description

Edge servers: An MEC system consists of M BSs denoted by $\mathcal{S} = \{s_1, ..., s_M\}$. The cell radius of BS $s_j \in \mathcal{S}$ is rad_j. The set of K subchannels in each BS s_j is denoted by $\mathcal{C}_j = \{c_j^1, ..., c_j^K\}$. We divide the total bandwidth B of each BS s_j equally into all subchannels $\mathcal{C}_j$. Each subchannel $c_j^k \in \mathcal{C}_j$ has a bandwidth $B_j^k = B/K$. Each BS s_j has an edge server installed, whose computing capacity is denoted by an $|\mathcal{R}|$-dimensional vector $R_j = (R_j^r)$ with dimension R_j^r being the capacity of resource type $r \in \mathcal{R} = \{\text{CPU, memory, storage,...}\}$.

Mobile users: This system's operational timeline is represented by a series of equal-length time slots t. Let $\mathcal{U}(t)$ denote the set of newly-arrived users u_i in time slot t, and $\mathcal{U}$ denote the set of all the current users in the system. We use an $|\mathcal{R}|$-dimensional vector $w_i = (w_i^r)$, $r \in \mathcal{R}$, to denote the required amount of computing resources to accommodate user u_i in an edge server. The distance from user u_i to BS s_j is $d_{j,i}$. We use $\mathcal{S}_i = \{s_j \in \mathcal{S} | d_{j,i} \leq rad_j\}, \forall u_i \in \mathcal{U}(t)$, to denote the set of user u_i's neighbor BSs, i.e., BSs that cover the user. The length of a user session (i.e., the period of time when a user uses the application, or is served by an edge server) is unknown at any time and represented by a number of time slots. For every user u_i, we need to make two decisions as follows.

User Allocation Decision. $\mathbf{a}_{j,i}^k(t)$ denotes user u_i's binary decision variable in time slot t. $\mathbf{a}_{j,i}^k(t) = 1$ if user u_i is assigned to BS s_j on subchannel c_j^k in time slot t; otherwise $\mathbf{a}_{j,i}^k(t) = 0$. Let $\mathbf{a}(t) = \{\mathbf{a}_i(t) | u_i \in \mathcal{U}(t)\}$ represent the user allocation strategy comprised of all the users' decisions in time slot t. $\mathbf{a}_i(t) \triangleq (s_j, c_j^k)$, where $\mathbf{a}_{j,i}^k(t) = 1$, indicates the BS and subchannel that serve user u_i in time slot t. We let $\mathbf{a}_i(t) \triangleq (0, 0)$ when user u_i is unallocated.

Power Allocation Decision. $\mathbf{p}_i(t)$ indicates user u_i's allocated transmit power in time slot t. Let $\mathbf{p}(t) = \{\mathbf{p}_i(t) | u_i \in \mathcal{U}\}$ represent the power allocation strategy comprised of the power allocation decisions for all the current users in the system in time slot t. We consider a discrete power control scheme [28], [29], [30], where the transmit power of each user is selected from a set $\mathcal{L}$ of discrete power levels. Discrete power control enables a simpler transmitter design than continuous power control and significantly reduces the overhead incurred by information exchange among network nodes [29].

Let $\mathcal{U}_j(t) = \{u_i \in \mathcal{U} | \sum_{k=1}^K \mathbf{a}_{j,i}^k(t) = 1\}$, be the set of users allocated to BS s_j in time slot t, and $\mathcal{U}_j^k(t) = \{u_i \in \mathcal{U} | \mathbf{a}_{j,i}^k(t) = 1\}$, be the set of users allocated to subchannel c_j^k in BS s_j in time slot t, $\mathcal{U}_j^k(t) \subseteq \mathcal{U}_j(t)$. A user cannot be allocated to multiple subchannels or BSs in a time slot. During a user session, an allocated user can switch to other subchannels in the same BS. We do not allow switching an allocated user to another BS unless necessary (a user might move outside the coverage area of its associated BS, in this case we consider the user as a new user in the next time slot) since it would interrupt the user session and require migrating user data back and forth from one edge server to another, which could be very costly if not impractical.

3.2 Allocation Delay Model

When user u_i is assigned to server s_j that has insufficient computing resources, u_i will be put in a waiting list Q_j waiting to be served. Once one or more existing users have left and free up the occupied resources, server s_j can start serving the users in the waiting list on a first-come, first-served basis. Given edge server s_j's computing capacity, we can easily calculate N_j, the maximum number of simultaneous users it can serve in a time slot. For instance, say there are three amounts/levels of computing resources w_i that might be required by each user u_i, $w_i \in \{< 3, 1, 2, 2 >, < 1, 3, 1, 2 >, < 2, 1, 3, 1 >\}$. An edge server s_j with a capacity of $< 43, 43, 41, 41 >$ is capable of serving 22 users simultaneously (7 users that require $< 3, 1, 2, 2 >$ each, 10 users that require $< 1, 3, 1, 2 >$ each, and 5 users that require $< 2, 1, 3, 1 >$ each); $N_j = 7 + 10 + 5 = 22$. We use ℓ to represent the expected length of a user session, which can be approximated based on historical data in practice. Edge server s_j's service rate can then be calculated by N_j/ℓ. Let $n_j(t)$ be the number of users that server s_j is serving and $Q_j(t)$ be the length of server s_j's waiting list in time slot t. The number of time slots that a newly-arrived user u_i has to wait until being served by s_j, or u_i's allocation delay cost, can be estimated by:

$$M_i(\mathbf{a}(t)) = \begin{cases} \frac{[n_j(t) - N_j + Q_j(t) + 1]_+}{N_j/\ell}, & \text{if} \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k(t) = 1 \\ M_{max}, & \text{if} \sum\limits_{s_j \in \mathcal{S}} \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k(t) = 0 \end{cases} \tag{1}$$

where M_{max} is the penalty when user u_i is unallocated, which can be any arbitrarily large number. An unallocated user always incurs a greater allocation delay cost than an allocated user to drive app vendors into allocating users. When user u_i is allocated to edge server s_j, its allocation delay cost is $\frac{[n_j(t) - N_j + Q_j(t) + 1]_+}{N_j/\ell}$. The denominator is the service rate of server s_j. Intuitively, the numerator ($[n_j(t) - N_j + Q_j(t) + 1]_+$) is the number of users currently served by s_j and users ahead of the newly-arrived user in the waiting list Q_j when s_j is exhausted of computing resources. We have $[n_j(t) - N_j + Q_j(t) + 1]_+ = 0$ when server s_j has enough resources to serve the new user immediately without putting it in waiting list Q_j. This allocation delay cost is one of the optimization objectives to be minimized.

3.3 Interference Model

3.3.1 Signal Model

With NOMA scheme, a BS transmits a superposition-coded signal to everyone on a subchannel [1]. In downlink transmissions, users employ SIC to decode the received superposed signal. Without loss of generality, suppose that all users $\mathcal{U}_j^k(t)$ on subchannel c_j^k are sorted by their channel conditions: $u_1, u_2, ..., u_{|\mathcal{U}_j^k(t)|}$, where u_1 has the weakest channel condition and $u_{|\mathcal{U}_j^k(t)|}$ has the strongest channel condition. User u_1, being the weakest user in $\mathcal{U}_j^k(t)$, decodes the signal without performing SIC. Then, user u_1's decoded component is subtracted from the superposed signal. The subsequent user in $\mathcal{U}_j^k(t)$, i.e., user u_2, can decode the received signal without interference from user u_1. Following this principle, the signal received by user $u_i \in \mathcal{U}_j^k(t)$ on subchannel c_j^k in BS s_j in time slot t has a signal-to-interference-plus-noise ratio $\gamma_i(t)$ of:

$$\gamma_i(t) = \frac{|h_{j,i}^k|^2 \mathbf{p}_i(t)}{|h_{j,i}^k|^2 \sum_{q=i+1}^{|\mathcal{U}_j^k(t)|} \mathbf{p}_q(t) + I_{j,i}^k(t) + \sigma^2} \tag{2}$$

where $|h_{j,i}^k|^2$ is user u_i's channel gain on subchannel c_j^k, $|h_{j,i}^k|^2 \sum_{q=i+1}^{|\mathcal{U}_j^k(t)|} \mathbf{p}_q(t)$ is the intra-cell interference experienced by user u_i (caused by those sharing the same subchannel with user u_i), $I_{j,i}^k(t) = \sum_{s_l \in \mathcal{S}_i \setminus \{s_j\}} |h_{l,i}^k|^2 p_l^k(t)$ is the inter-cell interference experienced by user u_i (caused by users in u_i's neighbor BSs), and σ^2 is the addictive white Gaussian noise. Taking into account the factors that affect a channel's condition, the SIC decoding order of users on subchannel c_j^k in time slot t must follow:

$$\Theta_j^k(t) \triangleq \frac{I_{j,1}^k(t) + \sigma^2}{|h_{j,1}^k|^2} \geq ... \geq \frac{I_{j,|\mathcal{U}_j^k(t)|}^k(t) + \sigma^2}{|h_{j,|\mathcal{U}_j^k(t)|}^k|^2} \tag{3}$$

where weaker users (high inter-cell interference and low channel gain) decode before stronger users. According to [14], $\Theta_j^k(t)$ is optimal for efficiently improving each individual user' data rate. When $\Theta_j^k(t)$ is followed, user u_i's achievable data rate r_i in time slot t is then:

$$r_i(t) = B_j^k \log_2\left(1 + \gamma_i(t)\right) \tag{4}$$

3.3.2 Interference Cost Model

Given allocation strategies $\mathbf{a}(t)$ and $\mathbf{p}(t)$, the interference-plus-noise $I_i(\mathbf{a}(t), \mathbf{p}(t))$ experienced by user u_i can be measured by:

$$I_i(\mathbf{a}(t), \mathbf{p}(t)) = \begin{cases} |h_{j,i}^k|^2 \sum_{q=i+1}^{|\mathcal{U}_j^k(t)|} \mathbf{p}_q(t) + I_{j,i}^k(t) + \sigma^2, & \\ \qquad\qquad \text{if } \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k(t) = 1 & \\ I_{max}, \qquad \text{if } \sum\limits_{s_j \in \mathcal{S}} \sum\limits_{c_j^k \in \mathcal{C}_j} \mathbf{a}_{j,i}^k(t) = 0 & \end{cases} \tag{5}$$

where I_{max} is the theoretical highest interference-plus-noise that a user may receive in any subchannel. I_{max} can also be any arbitrarily large number. It acts as a penalty for unallocated users to drive the allocation of users, similar to how the allocation delay cost is formulated in Section 3.2. This interference cost is one of the optimization objectives to be minimized. By minimizing the interference cost, the transmit power cost will consequently be minimized.

3.4 Problem Formulation

We formulate the online EUA problem as follows.

$$(\mathbf{P1}) \min \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{ \underbrace{\sum_{u_i \in \mathcal{U}} \left(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\right)}_{Y(\mathbf{a}(t), \mathbf{p}(t))} \right\}$$

$$\text{s.t.} \sum_{i=1}^{|\mathcal{U}_j|} \sum_{r=1}^{|\mathcal{R}|} \sum_{k=1}^{K} \mathbf{a}_{j,i}^k(t) w_i^r \leq R_j^r, \forall s_j \in \mathcal{S}, \forall t \tag{6a}$$

$$\sum_{j=1}^{M} \sum_{k=1}^{K} \mathbf{a}_{j,i}^k(t) d_{j,i} \leq rad_j, \forall u_i \in \mathcal{U}, \forall t \tag{6b}$$

$$\Theta_j^k(t), \forall s_j \in \mathcal{S}, \forall c_j^k \in \mathcal{C}_j, \forall t \tag{6c}$$

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{r_i(t)\right\} \geq r_{req}, \forall u_i \in \mathcal{U} \tag{6d}$$

$$\sum_{j=1}^{M} \sum_{k=1}^{K} \mathbf{a}_{j,i}^k(t) = 1, \forall u_i \in \mathcal{U} \tag{6e}$$

$$\sum_{k=1}^{K} \sum_{i=1}^{|\mathcal{U}|} \mathbf{a}_{j,i}^k(t) \mathbf{p}_i(t) \leq P_j, \forall s_j \in \mathcal{S}, \forall t \tag{6f}$$

$$\mathbf{a}_{j,i}^k(t) \in \{0, 1\}, \forall s_j \in \mathcal{S}, \forall u_i \in \mathcal{U}, \forall c_j^k \in \mathcal{C}_j \tag{6g}$$

$$\mathbf{p}_i(t) \in \mathbb{R}_{\geq 0}, \forall u_i \in \mathcal{U}, \forall t \tag{6h}$$

where η_1 and η_2 ($\eta_1 + \eta_2 = 1$) are the weights that indicate the importance of allocation delay and interference costs. Parameters η_1 and η_2 are adjustable and to be set empirically by the app vendor, depending on its priorities for allocation delay and interference cost (or energy efficiency). For example, if the app vendor wants to prioritize energy efficiency, it can increase the value of η_2, and vice versa. As the MEC system is highly stochastic, optimizing the long-term system performance is more beneficial than optimizing the short-term, spontaneous system performance. The objective is to minimize the time-average expectation of system cost, which consists of allocation delay costs and interference costs over multiple time slots. We use $Y(\mathbf{a}(t), \mathbf{p}(t)) = \sum_{u_i \in \mathcal{U}} \left(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\right)$ denote the system cost incurred by user and power allocation strategies $\mathbf{a}(t)$ and $\mathbf{p}(t)$ in time slot t.

Constraints (6a) and (6b) ensure that an edge server/BS does not accommodates users outside its computing capacity and cell coverage, respectively. Constraint (6c) enforces the SIC decoding order (3). Constraint (6d) ensures a long-term minimum data rate requirement r_{req} for every user. Constraint (6e) makes sure that a user is not allocated to multiple subchannels or BSs in a time slot. Constraint (6f) ensures that the total power assigned to all users in a BS does not exceed the BS's maximum power allowance at any time. Constraints (6g) and (6h) define the acceptable values of $\mathbf{a}_{j,i}^k(t)$ and $\mathbf{p}_i(t)$.

4 Online User and Power Allocation with Lyapunov Optimization

In this section, we introduce a Lyapunov optimization-based algorithm to solve problem **P1**.

4.1 Problem Transformation with Lyapunov Optimization

The MEC system being studied is a time-slotted system. The app vendor enforces a minimum data rate requirement r_{req} for its users. The conventional method would be to enforce this constraint in every time slot. In other words, every user would receive this exact data rate in every time slot. In contrast, our approach allows user data rate to be slightly higher or lower than the target data rate r_{req}, which might be more beneficial in terms of system cost minimization. In the long term, the data rate achieved by our approach still meets the data rate requirement (6d). As the user data rate can be slightly higher or lower than the target data rate in a time slot, there must be a mechanism to drive the data rate towards the target data rate in the next time slot(s), i.e., to stabilize users' average data rate over time. We introduce a concept called *accumulated data rate* for each user u_i, defined by:

$$D_i(t+1) = max\{D_i(t) + r_{req} - r_i(t), 0\} \tag{7}$$

where $D_i(0) = 0, \forall u_i \in \mathcal{U}$. The accumulated data rate $D_i(t+1)$ of a user u_i represents its overdue data rate accumulated over t time slots relative to the data rate requirement r_{req}. Its value increases if the user's data rate in the previous time slot $r_i(t)$ decreases and vice versa. This can be used to adjust the user and power allocation strategies to stabilize users' average data rate over time as enforced by (6d). To fulfill the long-term data rate requirement, $D_i(t)$ must be stabilized, or *mean rate stable* [31]: $\lim\limits_{t\to\infty} \frac{\mathbb{E}\{D_i(t)\}}{t} = 0$. Based on Eq. (7), we define a quadratic Lyapunov function: $L(D(t)) \triangleq \frac{1}{2}\sum_{u_i \in \mathcal{U}} D_i(t)^2$, where $D(t) \triangleq \{D_i(t), \forall u_i \in \mathcal{U}\}$. We can see that $L(D(t))$ is high when there is at least one user with a high accumulated data rate $D_i(t)$, and $L(D(t))$ is low when the accumulated data rate of every user is small, representing a stable state. We then define a *conditional Lyapunov drift* to observe how the Lyapunov function changes between two consecutive time slots: $\Delta(D(t)) \triangleq \mathbb{E}\{L(D(t+1)) - L(D(t)) | D(t)\}$. We minimize the system cost while stabilizing users' average data rate. By incorporating the system cost into the Lyapunov drift above, our optimization objective can be fulfilled without violating the data rate constraints. This can be achieved via a *drift-plus-penalty*:

$$\Delta(D(t)) + V\,\mathbb{E}\{\sum_{u_i \in \mathcal{U}} \big(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\big) | D(t)\}$$

where $V > 0$ is a parameter which adjusts the relative importance of the system cost to the accumulated user data rate. In other words, V controls the rate of stabilizing user data rate. Depending on the application context, app vendors can flexibly regulate the trade-off between time-average system cost and accumulated data rate by changing the value of V. For instance, they can increase V to relax the data rate requirement and put more emphasis on system cost minimization. Under the Lyapunov optimization scheme, we pursue the optimization objective in **P1** by minimizing the supreme bound of the above drift-plus-penalty.

Lemma 1. *Given any user and power allocation strategy in any time slot, the drift-plus-penalty is bounded by:*

$$\begin{aligned}
&\Delta(D(t)) + V\,\mathbb{E}\left\{\sum_{u_i \in \mathcal{U}} \big(\eta_1 M_i(\boldsymbol{a}(t)) + \eta_2 I_i(\boldsymbol{a}(t), \boldsymbol{p}(t))\big) | D(t)\right\} \\
&\leq O + \sum_{u_i \in \mathcal{U}} \mathbb{E}\left\{\frac{r_i^2(t)}{2} + D_i(t)\big(r_{req} - r_i(t)\big) - r_{req} r_i(t)\right. \\
&\qquad \left. + V\big(\eta_1 M_i(\boldsymbol{a}(t)) + \eta_2 I_i(\boldsymbol{a}(t), \boldsymbol{p}(t))\big) | D(t)\right\}
\end{aligned} \tag{8}$$

where $O = \frac{r_{req}^2}{2}$ is a finite constant.

Proof. See Appendix B. □

Next, we propose OUAD (Algorithm 1), an **O**nline **U**ser **A**llocation algorithm in **D**ynamic NOMA-based MEC systems, which formulates a user and power allocation strategy, $\mathbf{a}(t)$ and $\mathbf{p}(t)$, to lower the upper bound of the drift-plus-penalty (8) in every time slot, effectively solving Problem **P1**. Based on Lemma 1 and the concept of minimizing an expectation opportunistically [31], we can accomplish this by solving problem **P2** defined as follows in every time slot. **P2** is derived from the right-hand-side term of (8), which is the upper bound of the drift-plus-penalty.

$$\begin{aligned}
(\mathbf{P2}) \min_{\mathbf{a}(t), \mathbf{p}(t)} \sum_{u_i \in \mathcal{U}} \bigg(& \frac{r_i^2(t)}{2} + D_i(t)\big(r_{req} - r_i(t)\big) - r_{req} r_i(t) \\
& + V\big(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\big)\bigg)
\end{aligned} \tag{9}$$

$$\text{s.t. } (6a), (6b), (6c), (6e), (6f), (6g), (6h)$$

Algorithm 1 OUAD

```
1: Input: $\mathcal{S}, V, \eta_1, \eta_2, r_{req}$
2: Output: user and power allocation decisions $\mathbf{a}^k_{j,i}(t), \mathbf{p}_i(t), \forall t, \forall s_j \in \mathcal{S}, \forall u_i \in \mathcal{U}$
3: for every time slot $t$ do
4:     Observe newly-arrived users $\mathcal{U}(t)$ and each current user's accumulated data rate $D_i(t), \forall u_i \in \mathcal{U}$
5:     Determine $\mathbf{a}(t), \mathbf{p}(t)$ by solving P2
6:     Update users' accumulated date rate $D_i(t+1), \forall u_i \in \mathcal{U}$, according to Eq. (7)
7:     Update $Q_j(t+1), \forall s_j \in \mathcal{S}$
8: end for
```

In every time slot, newly-arrived users are assigned to BSs/edge servers with sufficient transmit power (Line 5 of Algorithm 1) based on the observed user arrivals and users' accumulated data rates (Line 4). Users that are assigned to an exhausted edge server will be put on a waiting list for that edge server and wait until one or more existing users depart. A user leaving the system releases computing resources, which can then be used to accommodate the waiting users. Users' accumulated data rates and waiting lists for all the edge servers are updated after a user and power allocation strategy has been determined in each time slot (Lines 6-7). OUAD works without prior knowledge of future user arrivals/departures, and statistics of the user distribution. This online algorithm allocates users as soon as they arrive and thus can accommodate user mobility and dynamic user sessions. When a user moves out of its associated BS's cell coverage, OUAD will treat it as a newly-arrived user and allocate it to another BS in the next time slot. The same treatment applies when an existing user terminates its current session and initiates a new one. Or if an application supports multiple concurrent user sessions by the same user, those sessions can be handled individually. The following theorem demonstrates that OUAD achieves an $[\mathcal{O}(1/V), \mathcal{O}(V)]$ trade-off between time-average system cost and accumulated data rate.

Theorem 1. *Let $y(t) \triangleq \sum_{u_i \in \mathcal{U}} \big(M_i(\boldsymbol{a}(t)) + I_i(\boldsymbol{a}(t), \boldsymbol{p}(t))\big)$, $y_{max} \triangleq max(y(t))$, and y_{opt} be the theoretical optimal solution of problem* **P2***. If there exists positive constants V, O, C, and ϵ*

such that for all possible values of $D(t)$ and all time slots t the following drift holds:

$$\Delta(D(t)) - V\,\mathbb{E}\left\{y(t)\middle|D(t)\right\} \leq O + C - \epsilon \sum_{u_i \in \mathcal{U}} D_i(t) - V y_{opt} \quad (10)$$

then the time-average system cost and accumulated data rate satisfy:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{y(t)\right\} \geq y_{opt} - \frac{O+C}{V} \quad (11)$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{u_i \in \mathcal{U}} \mathbb{E}\left\{D_i(t)\right\} \leq \frac{O + C + V(y_{max} - y_{opt})}{\epsilon} \quad (12)$$

Proof: See Appendix C of the supplementary file. □

This theorem implies that the time-average system cost and accumulated data rate are proportional to control parameter V. This provides app vendors with a powerful mechanism to control system performance. For example, V can be decreased for latency-sensitive services so that users' data rate can be quickly stabilized at the required minimum data rate. This theorem will be experimentally illustrated in Section 6.3.3.

Problem **P2** is NP-hard because its subproblem is already NP-hard, which we will show later. Problem **P2** is part of Problem **P1**, hence Problem **P1** is also NP-hard. Problem **P2** is hard to be solved optimally within a time slot. To find a near-optimal solution efficiently, we break it down into a user allocation problem (Stage #1 - Section 4.2) and a power allocation problem (Stage #2 - Section 4.3). We first assign users to subchannels in BSs in Stage #1. Here, we seek a solution with a low allocation delay cost that is most likely to result in low interference overall. We use the words "most likely" because we have not properly allocated transmit power to users yet, and thus the interference has not been evaluated or considered. Once all the users are allocated to BSs, we will proceed to Stage #2 to adjust their transmit power to meet their data rate requirements in an energy-efficient manner.

4.2 User Allocation Problem

In this phase, we allocate every user to a subchannel in a BS by solving problem **P3** modeled below.

$$(\mathbf{P3}) \min_{\mathbf{a}(t)} \sum_{u_i \in \mathcal{U}} \left(\frac{r_i^2(t)}{2} + D_i(t)\big(r_{req} - r_i(t)\big) - r_{req} r_i(t) + V\big(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\big) \right) \quad (13)$$

s.t. (6a), (6b), (6e), (6g)

Interference and transmit power are highly interdependent. An app vendor allocates transmit power to users based on the inter- and intra-cell interference they experience, which is partly caused by other users' transmit power. To handle this interdependence, we fix one decision (power allocation decision) while choosing the other (user allocation decision). At this stage, all the users are assigned a default transmit power. This allows us to estimate the interference experienced by users and incorporate it into optimization objective (13). This increases the possibility of low interference when a proper power allocation algorithm is applied later on. Problem **P3** is NP-hard because its special case is a reduction of the NP-complete PARTITION problem [32]. The proof employs the same technique used in Appendix B of [3] so it will be omitted here.

4.3 Power Allocation Problem

All the users are now assigned to subchannels in BSs once problem **P3** is solved. Next, we adjust their transmit power by solving problem **P4**, which is modeled by:

$$(\mathbf{P4}) \min_{\mathbf{p}(t)} \sum_{u_i \in \mathcal{U}} \left(\frac{r_i^2(t)}{2} + D_i(t)\big(r_{req} - r_i(t)\big) - r_{req} r_i(t) + V\eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t)) \right) \quad (14)$$

s.t. (6c), (6f), (6h)

$M_i(\mathbf{a}(t))$ is now excluded in the objective because power allocation decisions have no impact on users' allocation delays. The transmit power of users that arrived in previous time slots will also be adjusted because their received interference might change when there are new users arriving at the system.

5 User and Power Allocation Game

To effectively solve problem **P2** within each individual time slots, we first model it as a potential game. To find Nash equilibria in this game, we propose a decentralized two-stage algorithm, where each stage is responsible for solving a subproblem defined above (**P3** and **P4**).

5.1 Game Formulation and Properties

In each time slot t, an app vendor pursues objective (9) by finding a suitable user and power allocation strategy $\mathbf{a}(t)$ and $\mathbf{p}(t)$. The decisions $\mathbf{a}_i(t)$ and $\mathbf{p}_i(t)$ made for each individual user u_i is influenced by other users' decisions $\mathbf{a}_{-i}(t)$ and $\mathbf{p}_{-i}(t)$.

The EUA problem in a time slot is modelled as a game $Z = (\mathcal{U}(t), \{\mathcal{A}_i(t), \mathcal{P}_i(t)\}_{u_i \in \mathcal{U}(t)}, \{C(\mathbf{a}_i(t), \mathbf{p}_i(t))\}_{u_i \in \mathcal{U}})$, where $\mathcal{U}(t)$ is the set of players (users) arriving in time slot t, $\mathcal{A}_i(t)$ and $\mathcal{P}_i(t)$ are the sets of possible user and power allocation decisions available to each user u_i, and $C(\mathbf{a}_i(t), \mathbf{p}_i(t))$ is the cost incurred by decisions $\mathbf{a}_i(t)$ and $\mathbf{p}_i(t)$ made for user u_i (derived from Eq. (9)), the lower the better. Please note that cost $C(.)$ is not the same as the system cost $Y(\mathbf{a}(t), \mathbf{p}(t))$ formulated in Problem **P1**.

$$C(\mathbf{a}_i(t), \mathbf{p}_i(t)) = \sum_{u_i \in \mathcal{U}} \left(\frac{r_i^2(t)}{2} + D_i(t)\big(r_{req} - r_i(t)\big) - r_{req} r_i(t) + V\big(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\big) \right) \quad (15)$$

A Nash equilibrium of Z is a stable state of Z where an app vendor cannot lower the cost $C(.)$ any further by unilaterally changing the decision for any single user. In a Nash equilibrium $\mathbf{a}^*(t), \mathbf{p}^*(t)$, the allocation decision made for a user is the best reaction to the allocation decisions made for all other users [33]. This makes sure that if a Nash equilibrium exists, the decisions for all the users will automatically self-organize into a Nash equilibrium in finite iterations. In each iteration, every user proactively responds to the decisions made for all the other users to further reduce the cost $C(.)$. A potential game (defined below) always admits one or more Nash equilibria [34]. By showing that Z is a potential game, we can prove that it possesses at least one Nash equilibrium .

Definition 1. ***(Ordinal Potential Game)*** *An ordinal potential game is a game that has a potential function $\phi(\boldsymbol{a}_i(t), \boldsymbol{p}_i(t))$ satisfying $C(\boldsymbol{a}_i(t), \boldsymbol{p}_i(t)) > C(\boldsymbol{a}'_i(t), \boldsymbol{p}'_i(t)) \Leftrightarrow \phi(\boldsymbol{a}_i(t), \boldsymbol{p}_i(t)) > \phi(\boldsymbol{a}'_i(t), \boldsymbol{p}'_i(t))$, where $\boldsymbol{a}_i(t), \boldsymbol{a}'_i(t) \in \mathcal{A}_i(t)$, and $\boldsymbol{p}_i(t), \boldsymbol{p}'_i(t) \in \mathcal{P}_i(t)$.*

Clearly, Z is an ordinal potential game with cost function $C(.)$ being a direct potential function.

5.2 Algorithm Design

To find Nash equilibria in game Z, we present a distributed algorithm that adopts *best-response dynamics* [33], an iterative evolutionary procedure. In each iteration, we determine an allocation decision for every user by finding the best response to the decisions applied to other users. This decentralized procedure can be executed in parallel on edge servers, which coordinate the game through messaging synchronization [10], [35]. This procedure always converge to a Nash equilibrium thanks to *Finite Improvement Property* [34]. Our approach consists of two stages: Stage #1 (Algorithm 2) for solving problem **P3**, and Stage #2 (Algorithm 3) for solving problem **P4**.

Stage #1 (Algorithm 2): Algorithm 2 allocates every user to a subchannel in a BS. At this stage, every user is assigned a temporary default transmit power. Once Algorithm 2 completes, Algorithm 3 will adjust their transmit power to meet their data rate requirements. In Algorithm 2, all users are initially unallocated (Line 1). Subsequently, allocation decisions are updated and applied for every user iteratively (Lines 2-16), lowering the cost $C(.)$ after every iteration until it cannot be lowered any further - this is the Nash equilibrium. In each iteration, we determine the best decision $\mathbf{a}'_i(t)$ for each user u_i by iterating over all the subchannels in all of its neighbor BSs and select the subchannel that would generate the lowest cost $C(\mathbf{a}'_i(t), \mathbf{p}(t))$ if user u_i is to be allocated to it (Lines 3-10). If $C(\mathbf{a}'_i(t), \mathbf{p}(t))$ is not lower than the current cost $C(.)$, there is no need to update user u_i's current decision. If the new decision $\mathbf{a}'_i(t)$ leads to a lower cost $C(.)$, user u_i's current decision will be updated with $\mathbf{a}'_i(t)$. The request for applying $\mathbf{a}'_i(t)$ will be submitted for the opportunity to be officially applied (Lines 11-13). Among all the requests for decision applying, the one with the lowest cost $C(.)$ will be officially applied (Line 15). The user, whose request for decision update is selected, now has a new allocation decision. Note that the allocation strategy in an iteration is not final; it may be amended in following iterations if a new allocation decision for a user is found. Users assigned to an exhausted edge server will be put on a waiting list until one or more existing users depart the system and release the occupied computing resources.

The process of updating decisions for all users (Lines 4-14) can be executed in parallel because the processes for different users are independent of each other. The search for the best decision for each user (Lines 5-9) can also be parallelized. After all the users are allocated, we execute Stage #2 (Algorithm 3) to adjust their transmit powers.

Algorithm 2 Stage #1: User Allocation

```
    Input: $\mathcal{S}$, $\mathcal{U}(t)$, fixed power allocation strategy $\mathbf{p}(t)$
    Output: user allocation strategy $\mathbf{a}(t)$
 1: $\mathbf{a}_i(t) = (0, 0), \forall u_i \in \mathcal{U}(t)$
 2: repeat
 3:     Compute current cost $C(\mathbf{a}(t), \mathbf{p}(t))$
 4:     for each user $u_i \in \mathcal{U}(t)$ do
 5:         for each neighbor BS $s_j \in \mathcal{S}_i$ of user $u_i$ do
 6:             for each subchannel $c_j^k \in \mathcal{C}_j$ in BS $s_j$ do
 7:                 Compute $C(\mathbf{a}'_i(t), \mathbf{p}(t))$ – the new cost if
                    user $u_i$ is to be assigned to $c_j^k$
 8:             end for
 9:         end for
10:         Among all feasible decisions $\mathbf{a}'_i(t)$ above, find
            one that incurs the lowest cost $C(\mathbf{a}'_i(t), \mathbf{p}(t))$
11:         if $C(\mathbf{a}'_i(t), \mathbf{p}(t)) < C(\mathbf{a}(t), \mathbf{p}(t))$ then
12:             Request to apply $\mathbf{a}'_i(t)$
13:         end if
14:     end for
15:     Among all requests for applying decision update,
        apply the one with the lowest $C(\mathbf{a}'_i(t), \mathbf{p}(t))$
16: until decision updates not required for any users
17: Execute Stage #2
```

Stage #2 (Algorithm 3): Given the user allocation strategy found in Stage #1, Algorithm 3 adjusts the transmit power for all users. Note that we consider a discrete power control scheme, where the transmit power of a user is selected from a set $\mathcal{L}$ of discrete power levels. Initially, every user is allocated the lowest power level (Line 1). After that, for each user in each iteration, we iterate through every possible power level to find the one that incurs the lowest cost $C(\mathbf{a}(t), \mathbf{p}'_i(t))$ (Lines 5-8). If $C(\mathbf{a}(t), \mathbf{p}'_i(t))$ is lower than the cost incurred by the power allocation strategy found in the last iteration (calculated in Line 3), the request for updating this user's power will be submitted for a chance to be applied (Lines 9-11). Once all the users' requests for updating transmit power are submitted, the request with the lowest cost $C(.)$ will be officially applied in this iteration (Line 13). This iterative process terminates when we cannot lower the cost $C(.)$ by updating any user's transmit power.

5.3 Performance Analysis

5.3.1 Efficiency

Theorem 2 below assesses Algorithms 2 and 3's convergence time by the maximum number of iterations they may take before reaching a Nash equilibrium. In addition, their efficiency, or time complexity, is experimentally evaluated in Section 6.

Theorem 2. *The convergence time of Algorithms 2 and 3 is upper bounded by: $\frac{|\mathcal{U}|\left(\eta_1 M_{max}+\eta_2 I_{max}\right)}{min\{V\eta_1, Z\}}$ and $\frac{|\mathcal{U}|\left(\eta_1 M_{max}+\eta_2 I_{max}\right)}{X}$ iterations, respectively, where $Z = \frac{r_i'^2(t) - r_i''^2(t)}{2} + r_{req}(r''_i(t) - r'_i(t)) + V\eta_2|h_{min}|^2 p$, and $X = \frac{r_i'^2(t) - r_i''^2(t)}{2} + (D_i(t) + r_{req})(r''_i(t) - r'_i(t))$. Z and X represent the minimum decrease in cost $C(.)$ when the decision for a user is changed in the consequent iteration in Algorithm 2 and 3, respectively. Symbols*

Algorithm 3 Stage #2: Power Allocation

```
Input: $\mathcal{S}, \mathcal{U}$, user allocation strategy $\mathbf{a}(t)$ found in Stage #1, a set $\mathcal{L}$ of discrete power levels
Output: power allocation strategy $\mathbf{p}(t)$
Every user $u_i \in \mathcal{U}(t)$ is allocated the lowest power level
repeat
    Compute current cost $C(\mathbf{a}(t), \mathbf{p}(t))$
    for each user $u_i \in \mathcal{U}$ do
        for each power level $l \in \mathcal{L}$ do
            Compute $C(\mathbf{a}(t), \mathbf{p}'_i(t))$ – the new cost if user $u_i$ is given power level $l$
        end for
        Among all possible decisions $\mathbf{p}'_i(t)$ above, find one that incurs the lowest $C(\mathbf{a}(t), \mathbf{p}'_i(t))$
        if $C(\mathbf{a}(t), \mathbf{p}'_i(t)) < C(\mathbf{a}(t), \mathbf{p}(t))$ then
            Request to apply $\mathbf{p}'_i(t)$
        end if
    end for
    Among all requests for applying decision update, apply the one that has lowest $C(\mathbf{a}(t), \mathbf{p}'_i(t))$
until decision updates not required for any users
```

with a single apostrophe belong to one iteration and symbols with double apostrophes belong to the consequent iteration.

Proof: See Appendix D of the supplementary file. □

We also analyze the worst-case time complexity of Algorithms 2 and 3. The *sequential* time complexity of Algorithm 2 is $\mathcal{O}(G_{max} K M N_{ts}^3 \log N_{ts})$, where N_{ts} is the maximum number of new users in a time slot and G_{max} is the highest number of iterations (found by Theorem 2) that Algorithm 2 could take to find a Nash equilibrium. In every iteration, we find the best subchannel and BS for every new user by looking at all possible options (Lines 5-9 of Algorithm 2) to find an option that incurs the lowest cost $C(.)$ and submit it for the opportunity to be officially applied (Line 12 of Algorithm 2). For each user, there are at most K (subchannels) x M (BSs) options. In reality, the number of neighbor BSs of a user is considerably lower than M. In our simulations, each user has a maximum of three neighbor BSs. Each of the aforementioned options involves calculating the cost $C(.)$ for N_{ts} users, where each user costs $N_{ts} \log N_{ts}$. Calculating intra-cell interference dominates this calculation of $C(.)$ because it requires sorting users by their channel conditions (Eq. (3)). It is more computationally expensive than calculating inter-cell interference and allocation delay cost. Thus, Algorithm 2 costs $\mathcal{O}(G_{max} K M N_{ts}^3 \log N_{ts})$ if executed sequentially, which is rather computationally expensive. However, we can take advantage of the fact that the calculation of an option does not rely on the calculation of all other options. This allows Algorithm 2 to be run in parallel, lowering the complexity to roughly $\frac{\mathcal{O}(G_{max} K M N_{ts}^3 \log N_{ts})}{\mu}$, where μ is the number of processing threads in edge servers that run Algorithm 2 collectively. The experiments in Section 6 evaluate the efficiency of Algorithm 2 when running in parallel. Using the same reasoning, the parallel time complexity of Algorithm 3 is $\frac{\mathcal{O}(G_{max} |\mathcal{L}| N_{all}^3 \log N_{all})}{\mu}$, where N_{all} is the maximum number of current users in the system.

5.3.2 Effectiveness

We then analyze the theoretical optimality of the solutions found by Algorithms 2 and 3 by examining the Price of Anarchy (PoA) of the system cost $Y(.)$. PoA, being the ratio between the worst Nash equilibrium and the theoretical optimal strategy [34], is an important optimality/performance indicator for game theory-based methods [10], [35]. We use $(\mathbf{a}^{opt}(t), \mathbf{p}^{opt}(t))$ to denote the optimal strategy. The system-cost PoA is then defined as $\frac{\max\limits_{\mathbf{a}(t) \in \mathcal{A}(t), \mathbf{p}(t) \in \mathcal{A}(t)} Y(\mathbf{a}(t), \mathbf{p}(t))}{Y(\mathbf{a}^{opt}(t), \mathbf{p}^{opt}(t))}$.

Theorem 3. *The system-cost PoA in game Z satisfies:*

$$1 \leq PoA \leq \frac{\eta_1 \frac{[|\mathcal{U}| - N_{min}]_+}{N_{min}/\ell} + \eta_2(|h_{max}|^2 l_{max} + \sigma^2)}{\eta_2 \sigma^2} \tag{16}$$

where N_{min} is the minimum service rate (which belongs to the most resource-limited edge server), $|h_{max}|^2$ is a user's highest possible channel gain, and $l_{max} \in \mathcal{L}$ is a user's highest power level.

Proof: See Appendix E of the supplementary file. □

This theorem reaffirms that we can improve the performance of the Nash equilibria, or decrease the gap between the worst Nash equilibrium and the optimal solution, when users' channel gains ($|h_{max}|^2$) are low, when the number of users ($|\mathcal{U}|$) is low, or by increasing edge servers' service rates N_{min} (which can be achieved by increasing edge servers' capacities).

6 Performance Evaluation

6.1 Performance Benchmark

In our experiments, we evaluate OUAD against four representative approaches:

- *SUAC* [24]: This online user allocation approach minimizes the allocation delay cost in a time-slotted setting. It does not incorporate a proper power control scheme. Thus, to ensure the fairness of the comparison, after all users are allocated to BSs, we execute DPC-SPM, a state-of-the-art power control algorithm in NOMA [18], to assign minimum transmit power to users while meeting their data rate requirements.
- *SCG-SA* [36]: This approach aims to increase the energy efficiency while meeting user data rate requirements in NOMA-based cellular networks. SCG-SA only allocates users, who are presumably already allocated to BSs, to subchannels. It does not allocate users to BSs. Thus, we will allocate users to their nearest BSs. Subsequently, SCG-SA allocates users to subchannels based on a ranking of user channel conditions. This approach is designed without the consideration of time-slotted scenarios. In the experiments, we execute this approach once in every time slot. Unallocated users in a time slot are considered as new users in the next time slot.
- *miUA* [3]: This approach incorporates both inter- and intra-cell interference in NOMA-based MEC networks and also aims to maximize the energy efficiency. Similar to SCG-SA, miUA is designed without the consideration of time-slotted scenarios so we will execute this approach in every time slot. Unallocated

TABLE 1: Experiment settings

BS maximum transmit power (P_j)	46dBm
Inter-site distance	500m
Cell radius (R_j)	289m
Minimum distance between user & BS	35m
Thermal noise density	-174dBm/Hz
Large-scale propagation model	$128.1 + 37.6log_{10}(d_{j,i})$dB
System bandwidth (B)	10MHz

TABLE 2: Experiment sets

	Traffic intensity ζ	Data rate requirement r_{req}	Control parameter V
Set #1	$0.04, 0.041, ..., 0.049$	0.5	5
Set #2	0.045	$0.2, 0.3, ..., 0.7$	5
Set #3	0.045	$0.3, 0.5, 0.7$	$1, 2, ..., 10$

users in a time slot are considered as new users in the next time slot.

- *Join Shortest Queue (JSQ)*: In each time slot, each user is allocated to the neighbor BS/server which has the shortest waiting list. This approach employs DPC-SPM power allocation method.

6.2 Experiment Settings

Our experiments are compliant with the LTE specifications [37] (Table 1). We consider a 7-cell hexagon-layout network, corresponding to 7 BSs/edge servers, each with 6 communication subchannels. Each user requires four types of computing resources, $\mathcal{R} = \{\text{CPU}, \text{storage}, \text{RAM}, \text{GPU}\}$. We randomly generate edge servers' computing capacities using a normal distribution $\mathcal{N}(80,\, 20^2)$, where 80 is the average amount of each computing resource, and 20 is the standard deviation. All edge servers combined can serve $N = \sum_{s_j \in \mathcal{S}} N_j$ concurrent users. User arrivals are generated based on a Poisson process at rate $[0, \zeta N]$, where $\zeta \in \mathbb{R}$ controls the traffic intensity. They are uniformly distributed within BSs' coverage areas. w_i is randomly picked from three normalized levels $\{$<2,2,3,3>,<2,2,2,1>,<1,1,1,2>$\}$. The set $\mathcal{L}$ of discrete transmit power levels is set to $\{-30\text{dBm}, -29\text{dBm}, ..., 23\text{dBm}\}$. Each user session's length is uniformly drawn from 10 to 20 1-second time slots. We conduct three sets of experiments (summarized in Table 2).

6.3 Experimental Results

6.3.1 Impact of Traffic Intensity (Set #1)

In Set #1, we simulate different user arrival rates by varying the traffic intensity ζ. When ζ increases, the allocation delay experienced by a user on average (Fig. 2) gradually increases because of the rising number of users joining the system (for reference, there are around 20 new users in each time slot when $\zeta = 0.049$). SUAC, whose sole objective is to minimize the allocation delay, clearly achieves the lowest allocation delay among all the approaches, closely followed by JSQ and OUAD. miUA achieves the worst performance with an average allocation delay twice higher than OUAD.

The energy efficiency (Fig. 3) is measured by the ratio of the total data rate to the total power consumption. miUA is the most energy-efficient method since it solely focuses on minimizing inter- and intra-cell interference. However, its energy efficiency comes at the price of very long allocation delays (Fig. 2). OUAD achieves a much lower allocation delay while its energy efficiency is only slightly lower than miUA (even on par with miUA in some cases). OUAD and miUA are remarkably more energy-efficient than the other three approaches. This shows the significance of considering interference in user allocation. Fig. 4 depicts the total transmit power required by all the approaches. We can see that SCG-SA, SUAC, and JSQ consume more power than OUAD and miUA by orders of magnitude. Fig. 5 illustrates the cumulative distribution function (CDF) of users' average data rate. A great portion of users allocated by SCG-SA, SUAC, and JSQ achieves either very low or very high data rates, largely deviating from the target data rate $r_{req} = 0.5$Mbps. The average data rate of users allocated by OUAD is slightly higher than those allocated by miUA.

Fig. 6 visualizes the time efficiency under different traffic intensities. The line plot shows the average elapsed CPU time per time slot. The bar plot shows the number of decision iterations taken by Algorithms 2 and 3, which is a commonly used efficiency metric for game-theoretic approaches [10], [38] because of its machine independence (the time taken to solve a problem varies machine to machine). When traffic intensity ζ increases, Algorithms 2 and 3 require more iterations, consequently higher CPU time in total. OUAD is slightly faster than miUA. OUAD and miUA are the slowest due to the complexity of calculating user interference. Nevertheless, their completion time is still well within an acceptable range. In each time slot, OUAD allocates all new users within around 30ms – well below the duration of each time slot (1 second). This ensures that OUAD can be practically applied in MEC systems where low latency is mandated. In the event where OUAD takes longer than a time slot to reach a Nash equilibrium, which we anticipate to be very rare, OUAD can continue and take this time slot as a slightly longer time slot. The timeline of the system resumes when OUAD finishes in this abnormal time slot. This will have no impact on the operation of OUAD.

6.3.2 Impact of Minimum Data Rate Requirement (Set #2)

In Set #2, we vary the minimum data rate requirement r_{req}. Unsurprisingly, this does not have any impact on the allocation delay as shown in Fig. 7, which remains unchanged regardless of the changing r_{req}. Again, OUAD achieves a very low allocation delay, only marginally higher than SUAC, whose only goal is to lower the allocation delay cost. miUA is the most energy-efficient method (Fig. 8) at the cost of very high allocation delays (Fig. 7). OUAD's energy efficiency is very close to miUA, while its average allocation delay is much lower than miUA. In general, when r_{req} increases, all the approaches require more transmit power to serve users (Fig. 9). Their energy efficiency decreases accordingly. Fig. 10 depicts the CDF of the average data rate of all users. Again, contrasted to OUAD and miUA, SCG-SA, SUAC, and JSQ largely deviate from the minimum data rate requirement $r_{req} = 0.7$Mbps. They fail to deliver satisfactory data rates to a great number of users and meanwhile, they provide excessive data rates to an equally great number of users. This demonstrates an extremely inefficient use of transmit power.

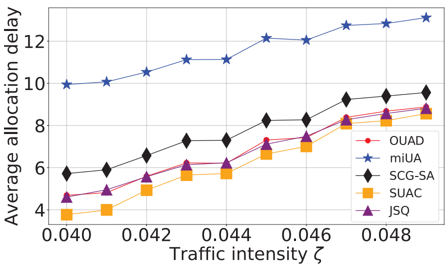

Fig. 2: Average allocation delay vs. traffic intensity ζ (Set #1).

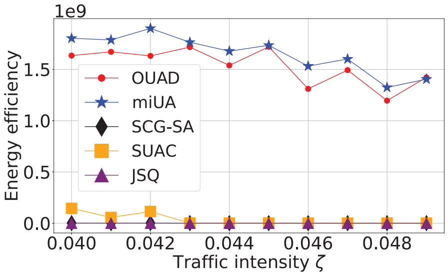

Fig. 3: Energy efficiency vs. traffic intensity ζ (Set #1).

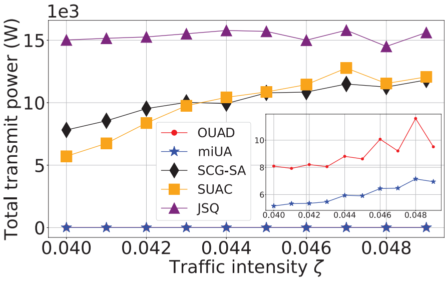

Fig. 4: Total transmit power vs. traffic intensity ζ (Set #1).

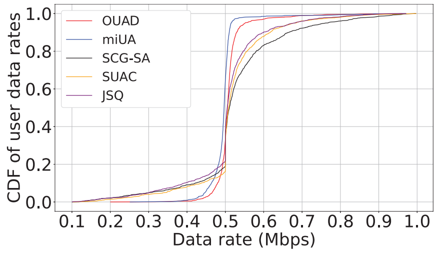

Fig. 5: CDF of all the users' data rates (Set #1, $V = 0.045$).

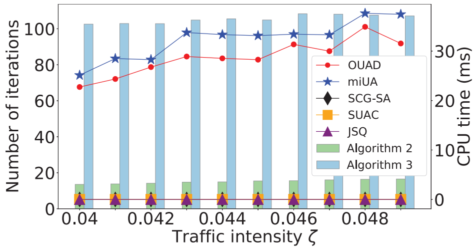

Fig. 6: Average number of decision iterations and elapsed CPU time per time slot vs. traffic intensity ζ (Set #1).

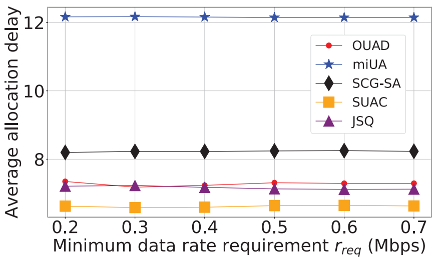

Fig. 7: Average allocation delay vs. data rate requirement r_{req} (Set #2).

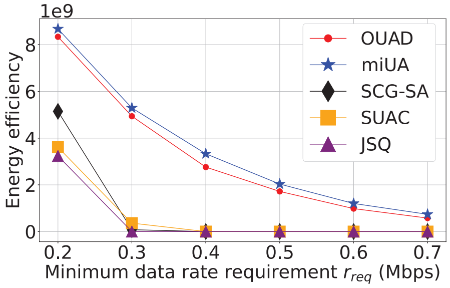

Fig. 8: Total transmit power vs. data rate requirement r_{req} (Set #2).

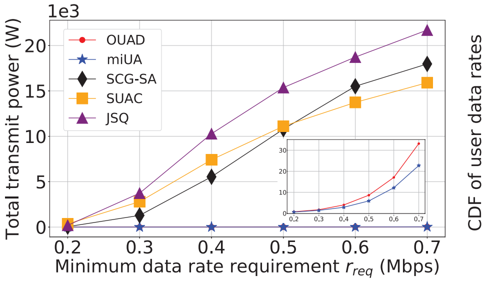

Fig. 9: Total transmit power vs. data rate requirement r_{req} (Set #2).

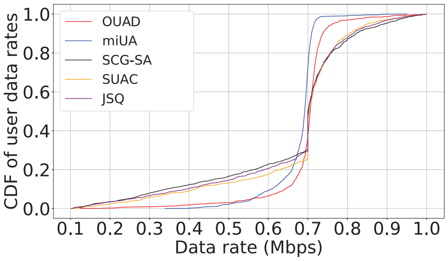

Fig. 10: CDF of all the users' data rates (Set #2, $r_{req} = 0.7$Mbps).

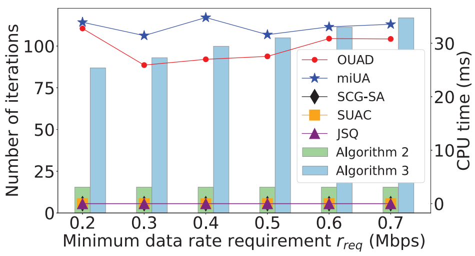

Fig. 11: Average number of decision iterations and elapsed CPU time per time slot vs. data rate requirement r_{req} (Set #2).

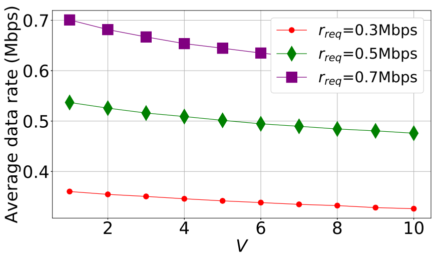

Fig. 12: Control parameter V vs. users' average data rate by OUAD (Set #3).

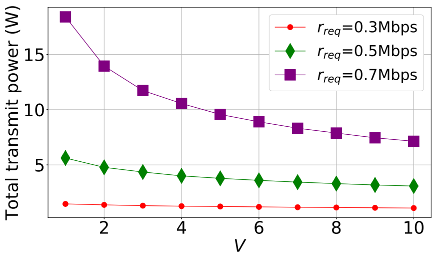

Fig. 13: Control parameter V vs. total transmit power by OUAD (Set #3).

Fig. 11 shows the time efficiency. Changing r_{req} does not affect Algorithm 2, thus its number of decision iterations remains the same in all settings. The change in r_{req} impacts only Algorithm 3. An increase in r_{req} increases the complexity of Algorithm 3, which now requires more iterations to converge to a Nash equilibrium. This leads to a slight increase in CPU time. Again, in each time slot, OUAD takes roughly 30ms to allocate all new users, which is within an acceptable range and well below the length of a time slot.

6.3.3 Impact of Parameter V (Set #3)

We examine how parameter V impacts the user average data rate achieved by OUAD under different data rate requirements r_{req}. Based on (9), a greater value of V results in a lower emphasis on the accumulated data rate (7). An increase in V lowers users' average data rate (Fig. 12), and consequently decreases the total transmit power required (Fig. 13). When V is very high, users' average data rate is even below the data rate requirement r_{req} since it now takes longer for users' data rates to converge to r_{req}. Because of the slow convergence, many users already left the system before their data rates can converge to a satisfactory level. This observation shows that app vendors can adjust V to flexibly control the trade-off between users' average data rate and energy consumption or system cost depending on their app-specific requirements.

6.3.4 Discussion

The three experiment sets have demonstrated the performance, flexibility, and practicality of the proposed approach OUAD. We aim to select the simulation parameters that are the most representative in any scenario.

The first parameter is traffic intensity ζ, or the number of users arriving in each time slot (Experiment Set #1), which is an important parameter in any highly stochastic MEC system. OUAD is highly efficient even under the highest traffic intensity.

Unlike the traffic intensity which app vendors have no control of, the next parameter, minimum data rate requirement r_{req} (Experiment Set #2), can be adjusted by app vendors. Varying r_{req} has no impact on the allocation delay and thus allows us to study its impact on user satisfaction or QoE at a fine granularity. A higher r_{req} is more resource-demanding so more users would not receive a satisfactory service or date rate without a proper user and power allocation like OUAD. Parameter r_{req} also affects the energy efficiency of the system.

The last parameter, V (Experiment Set #3), can be adjusted by app vendors to control the trade-off between users' average data rate and energy consumption or system cost. This experiment demonstrates the flexibility of OUAD in allowing app vendors to adapt to different settings that are specific to different applications and environments.

In summary, OUAD outperforms all other approaches in all experimental settings. It is efficient and can effectively allocate transmit power to users and users to edge servers/BSs, achieving great energy efficiency while getting satisfactory data rates for the most users. OUAD can function in multi-cell multi-channel environments and does not assume a limit on the number of users on each subchannel.

7 Related Work

MEC changes the provision of computing and storage resources structurally and raises many new problems, e.g., edge user allocation [35], edge data caching [39], edge data integrity [40], edge DDoS mitigation [41], collaborative edge computing [42], [43], hierarchical edge intelligence [44], etc.

The edge user allocation (EUA) problem has been extensively studied recently [2], [4], [5], [6], [7], [11], [12], [13], [24], [35], [45], [46], [47]. The authors of [7], [13] aim to assign as many users to as few edge servers as possible. This objective is often unrealistic in highly stochastic MEC systems since app vendors need to utilize as many edge servers as possible to serve their users. In [12], the authors incorporate user mobility into the user allocation and aim to increase user coverage rate and decrease the number of reallocations, which might occur when a user move between edge servers. The authors of [4], [11], [35] minimize the system cost calculated by how much computing resources are required to accommodate users. In [5], [6], [47], the authors solve an EUA problem where an app vendor can dynamically adjust the QoS levels of its users, which corresponds to the users' resource consumption. Among the aforementioned studies, only [4], [11] incorporate the communication aspect of MEC, e.g., the availability of multiple communication subchannels and wireless interference. Nonetheless, they do not incorporate inter-cell interference, which impacts users' data rates significantly and must not be neglected in dense 5G/6G networks. This is highly uneconomical since app vendors can now access and leverage network data such as received signal, received power, throughput, neighbor cells, QoS, etc. [8], [9]. More importantly, none of those studies considers NOMA (the de facto 5G/6G multi-access scheme), where power control and interference must not be ignored. The authors of [3] allocate users in power-domain NOMA-based MEC systems so that the transmit power cost is minimized. They incorporate both inter- and intra-cell interference. To allocate transmit power, they use an existing state-of-the-art power allocation method (DPC-SPM [18]), which is not suitable in our problem because it requires a specific data rate, i.e., every user would receive this data rate in every time slot. Our approach allows user data rate to be slightly higher or lower than the target data rate, which might be more beneficial in terms of system cost minimization. In the long term, the data rate achieved by OUAD still meets the data rate requirement. Furthermore, [3] and most of the aforementioned studies do not take into account the temporal dimension of MEC systems in which users come and go over time. Without it, the allocation delay experienced by users could be very high and their QoE would be impacted profoundly. The authors of [24] consider the temporal dimension and allocation delay but completely ignore the communication aspect of MEC. The authors of [48] study the EUA problem in an online setting where users come and go over time. However, they do not consider the scenario where users might have to wait to be served and incur allocation delay costs. Their approach may leave users unallocated.

Computation offloading is a very popular problem in MEC that share many similarities with the EUA problem. However, they are distinct from each other by several essential characteristics. In the EUA problem, a user can only be assigned to one edge server. To serve the user, the edge server must dedicate computing resources (CPU cores, memory, storage, etc.), which must always be available during the user session. For example, a machine learning application would require storage to store data, which is then loaded into the memory before being processed by CPUs or GPUs. While in the computation offloading problem, each user generates a series of computation tasks, which can be processed on multiple edge servers (partial/full offloading) and/or their local device [49]; and a task is usually single-dimensional (measured by CPU frequency) [22], [23], [50], [51], [52], [53]. Computation offloading problems usually concern with task execution/computation delay [22], [23], [51], [52], [54], which are not a concern for the EUA problem.

In conventional cellular networks, the user allocation problem (a.k.a. user association/clustering or BS assignment problem) is a mature problem [14], [15], [16], [17], [18], [19], [20], [55]. Since the introduction of NOMA, this line of research has again received tremendous attention. However, many of them make unrealistic assumptions, for instance, a limit on the number of users on each subchannel [18], [19], [20], and single-cell [16], [17] or single-channel [14], [15] systems. These assumptions unnecessarily impede the prospects of NOMA and render their approaches impractical in real-world NOMA-powered MEC systems. The works in [48], [56], [57] aim to maximize the sum-rate, which consequently result in energy inefficiency. Whereas in most real-world scenarios, users only require a specific level of data rate to ensure the QoE. In [36], [46], the authors aim to allocate users so that the energy efficiency is maximized. However, they, and other existing user allocation approaches, are unsuitable in MEC environments because they do not consider the temporal dimension and computation aspect of MEC like we do.

8 Conclusion and Future Work

We tackle an online user allocation problem in multi-channel multi-cell mobile edge computing (MEC) systems powered by power-domain non-orthogonal multiple access (NOMA). The temporal dimension is incorporated to accommodate users randomly arriving and departing the MEC system over time. Our aim is to minimize the costs of allocation delay and transmit power, improving the energy efficiency while satisfying several constraints in NOMA-based MEC systems, including a long-term user data rate constraint. We propose OUAD, a Lyapunov-based online user allocation algorithm that allocates users without any data of future user arrivals and departures. OUAD decomposes a long-term problem into a series of subproblems to be solved in individual time slots. To effectively and efficiently solve the subproblem in each time slot, OUAD employs a game-theoretical approach, which is evaluated theoretically. OUAD is shown to significantly outperform all the representative approaches through a series of experiments.

User allocation in NOMA-based MEC will require more attention as NOMA continues to advance. App vendors' access to network information (received signal, received power, throughput, neighbor cells, etc.) in MEC systems will raise new security concerns. Beside downlink transmission, uplink transmission also needs to be investigated for applications that receive a great amount of data from users. We will also attempt to validate the practicality of OUAD further by 1) tightening up its performance bounds; and 2) evaluate its performance on a real-world testbed.

Acknowledgments

This work is supported by Laureate Fellowship FL190100035, and Australian Research Council Discovery Project grants DP200102491, DP180100212, and DP170101932.

References

[1] 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.

[2] E. Liu, L. Zheng, Q. He, B. Xu, and G. Zhang, "Criticality-awareness edge user allocation for public safety," *IEEE Transactions on Services Computing*, 2021.

[3] P. Lai, Q. He, G. Cui, F. Chen, J. Grundy, M. Abdelrazek, J. G. Hosking, and Y. Yang, "Cost-Effective User Allocation in 5G NOMA-based Mobile Edge Computing Systems," *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3077470.

[4] G. Cui, Q. He, F. Chen, Y. Zhang, H. Jin, and Y. Yang, "Interference-aware game-theoretic device allocation for mobile edge computing," *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3064063.

[5] S. P. Panda, K. Ray, and A. Banerjee, "Dynamic edge user allocation with user specified qos preferences," in *Proceedings of International Conference on Service-Oriented Computing*. Springer, 2020, pp. 187–197.

[6] P. Lai, Q. He, G. Cui, F. Chen, M. Abdelrazek, J. Grundy, J. Hosking, and Y. Yang, "Quality of experience-aware user allocation in edge computing systems: A potential game," in *Proceedings of International Conference on Distributed Computing Systems*. IEEE, 2020, doi: 10.1109/ICDCS47774.2020.00036.

[7] 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.

[8] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "MEC in 5G networks," *ETSI white paper*, vol. 28, pp. 1–28, 2018. [Online]. Available: www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[9] D. Sabella, V. Sukhomlinov, L. Trang, S. Kekki, P. Paglierani, R. Rossbach, X. Li, Y. Fang, D. Druta, F. Giust *et al.*, "Developing software for multi-access edge computing," *ETSI white paper*, vol. 20, pp. 1–38, 2019. [Online]. Available: ww.etsi.org/images/files/ETSIWhitePapers/etsi_wp20ed2_MEC_SoftwareDevelopment.pdf

[10] 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.

[11] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between multi-tenancy and interference: A service user allocation game," *IEEE Transactions on Services Computing*, 2020, doi: 10.1109/TSC.2020.3028760.

[12] 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.

[13] 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*, 2020, doi: 10.1109/TCC.2020.3001570.

[14] 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.

[15] X. Zhang, Q. Gao, C. Gong, and Z. Xu, "User grouping and power allocation for noma visible light communication multi-cell networks," *IEEE communications letters*, vol. 21, no. 4, pp. 777–780, 2016.

[16] M. A. Sedaghat and R. R. Müller, "On user pairing in uplink NOMA," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3474–3486, 2018.

[17] 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.

[18] 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.

[19] L. You, D. Yuan, L. Lei, S. Sun, S. Chatzinotas, and B. Ottersten, "Resource optimization with load coupling in multi-cell noma," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4735–4749, 2018.

[20] 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.

[21] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5g ultra-dense cellular networks," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 72–79, 2016.

[22] 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.

[23] 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.

[24] P. Lai, Q. He, X. Xia, F. Chen, M. Abdelrazek, J. Grundy, J. G. Hosking, and Y. Yang, "Dynamic user allocation in stochastic mobile edge computing systems," *IEEE Transactions on Services Computing*, 2021, doi: 10.1109/10.1109/TSC.2021.3063148.

[25] T. Atchley, "Queue times for amazon's new world mmo have been off the chart," visited 8-Dec-2021. [Online]. Available: www.blizzardwatch.com/2021/09/29/new-world-queue-time/

[26] J. Carvalho, "Diablo 2: Resurrected queue times and server issues are out of control," visited 8-Dec-2021. [Online]. Available: www.gamerant.com/diablo-2-resurrected-queue-times-server-issues-out-of-control/

[27] J. Moore, "Final fantasy 14: Endwalker players are queuing to play... then getting error codes," visited 8-Dec-2021. [Online]. Available: www.ign.com/articles/final-fantasy-14-endwalker-queuing-error-codes

[28] L. Lei, D. Yuan, C. K. Ho, and S. Sun, "Power and channel allocation for non-orthogonal multiple access in 5g systems: Tractability and computation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8580–8594, 2016.

[29] H. Zhang, L. Venturino, N. Prasad, P. Li, S. Rangarajan, and X. Wang, "Weighted sum-rate maximization in multi-cell networks via coordinated scheduling and discrete power control," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 6, pp. 1214–1224, 2011.

[30] W.-J. Huang, Y.-W. Hong, and C.-C. J. Kuo, "Discrete power allocation for lifetime maximization in cooperative networks," in *2007 IEEE 66th Vehicular Technology Conference*. IEEE, 2007, pp. 581–585.

[31] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

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

[33] M. J. Osborne *et al.*, *An introduction to game theory*. Oxford university press New York, 2004, vol. 3, no. 3.

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

[35] 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.

[36] 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.

[37] 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.

[38] J. Zhang, P. Lu, Z. Li, and J. Gan, "Distributed trip selection game for public bike system with crowdsourcing," in *Proceedings of INFOCOM*. IEEE, 2018, pp. 2717–2725.

[39] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, "Online collaborative data caching in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 281–294, 2021.

[40] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Auditing cache data integrity in the edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1210–1223, 2021.

[41] Q. He, C. Wang, G. Cui, B. Li, R. Zhou, Q. Zhou, Y. Xiang, H. Jin, and Y. Yang, "A game-theoretical approach for mitigating edge DDoS attack," *IEEE Transactions on Dependable and Secure Computing*, 2021. [Online]. Available: http://dx.doi.org/10.1109/TDSC.2021.3055559

[42] L. Yuan, Q. He, S. Tan, B. Li, J. Yu, F. Chen, H. Jin, and Y. Yang, "CoopEdge: A decentralized blockchain-based platform for cooperative edge computing," in *The Web Conference*, 2021, pp. 2245–2257.

[43] L. Yuan, Q. He, F. Chen, J. Zhang, L. Qi, X. Xu, Y. Xiang, and Y. Yang, "CSEdge: Enabling collaborative edge storage for multi-access edge computing based on blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1873–1887, 2021.

[44] Q. He, Z. Dong, F. Chen, S. Deng, W. Liang, and Y. Yang, "Pyramid: enabling hierarchical neural networks with edge computing," in *The Web Conference*, 2022, pp. 1860—-1870. [Online]. Available: http://dx.doi.org/10.1145/3485447.3511990

[45] C. Wu, Q. Peng, Y. Xia, Y. Ma, W. Zheng, H. Xie, S. Pang, F. Li, X. Fu, X. Li *et al.*, "Online user allocation in mobile edge computing environments: A decentralized reactive approach," *Journal of Systems Architecture*, vol. 113, p. 101904, 2021.

[46] X. Liu, H. Zhang, K. Long, A. Nallanathan, and V. C. Leung, "Energy efficient user association, resource allocation and caching deployment in fog radio access networks," *IEEE Transactions on Vehicular Technology*, 2021, doi: 10.1109/TVT.2021.3131720.

[47] P. Lai, Q. He, G. Cui, X. Xia, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Qoe-aware user allocation in edge computing systems with dynamic qos," *Future Generation Computer Systems*, vol. 112, pp. 684–694, 2020.

[48] G. Cui, Q. He, X. Xia, F. Chen, F. Dong, H. Jin, and Y. Yang, "Ol-eua: Online user allocation for noma-based mobile edge computing," *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3112941.

[49] 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.

[50] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 771–786, 2018.

[51] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 244–12 258, 2018.

[52] C. Xu, G. Zheng, and L. Tang, "Energy-aware user association for noma-based mobile edge computing using matching-coalition game," *IEEE Access*, vol. 8, pp. 61 943–61 955, 2020.

[53] 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.

[54] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks," *IEEE Journal on Selected Areas in Communications*, 2020, doi: 10.1109/JSAC.2020.3018809.

[55] J. Zheng, Y. Cai, Y. Liu, Y. Xu, B. Duan, and X. Shen, "Optimal power allocation and user scheduling in multicell networks: Base station cooperation using a game-theoretic approach," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 6928–6942, 2014.

[56] 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.

[57] L. Salaün, M. Coupechoux, and C. S. Chen, "Weighted sum-rate maximization in multi-carrier noma with cellular power constraint," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 451–459.

Phu Lai received his PhD degree in Information and Communications Technology from Swinburne University of Technology, Australia, in 2021. He is currently a post-doc research fellow at Cisco-La Trobe Centre for AI and Internet of Things, La Trobe University, Australia. His research interests include IoT, software engineering, and cloud/edge computing.

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

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.

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.

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/.

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.

Appendix A
Key Notations

Key notations used in this paper are listed in Table 3.

TABLE 3: Key Notations

Symbol	Description
$\mathbf{a}^k_{j,i}(t)$	user allocation decision for user u_i. Its value is 1 if the user is to be allocated to the k-th subchannel of BS s_j in time slot t. Otherwise, its value is 0
B^k_j	bandwidth of channel c^k_j in BS s_j
$\mathcal{C}_j$	the set of subchannels $c^k_j, k \in \{1, 2, ..., K\}$, in BS s_j
$C(\mathbf{a}_i(t), \mathbf{p}_i(t))$	the cost incurred by a user allocation strategy $\mathbf{a}_i(t)$ and a power allocation strategy $\mathbf{p}_i(t)$ in time slot t.
$d_{j,i}$	distance between user u_i and BS s_j
$D_i(t)$	the accumulated data rate of user u_i in time slot t
$\|h^k_{j,i}\|^2$	channel gain of user u_i on subchannel c^k_j in BS s_j
$I^k_{j,i}(t)$	inter-cell interference experienced by user u_i on subchannel c^k_j in time slot t
$I_i(\mathbf{a}(t), \mathbf{p}(t))$	the interference-plus-noise cost of user u_i given a user allocation strategy $\mathbf{a}(t)$ and a power allocation strategy $\mathbf{p}(t)$ in time slot t
I_{max}	the penalty on interference cost of an unallocated user
ℓ	expected user session length
$M_i(\mathbf{a}(t))$	the allocation delay cost of user u_i given a user allocation strategy $\mathbf{a}(t)$ in time slot t
M_{max}	the penalty on allocation delay cost of an unallocated user
$n_j(t)$	the number of users being served by edge server s_j in time slot t
N_j	the maximum number of concurrent users in edge server s_j
P_j	maximum transmit power (power capacity) of BS s_j
$\mathbf{p}_i(t)$	allocation decision on the amount of transmit power allocated to user u_i in time slot t
$p^k_j(t)$	total transmit power of BS s_j on subchannel c^k_j in time slot t
$Q_j(t)$	the number of users waiting to be served by edge server s_j in time slot t
R_j	computing capacity of edge server s_j. R_j is a $\|\mathcal{R}\|$-dimensional vector
rad_j	cell radius of BS s_j
$r_i(t)$	achievable data rate of user u_i in time slot t
$\mathcal{S}$	the set of BS/edge servers $s_j, j \in \{1, 2, ..., M\}$
$\mathcal{S}_i$	set of user u_i's neighbor BSs
$\mathcal{R}$	the set of computing resource types, or computing capacity dimensions. $\mathcal{R} = \{\text{CPU}, \text{RAM}, \text{storage}, ...\}$
$\mathcal{U}(t)$	the set of incoming users in time slot t
$\mathcal{U}$	the set of all current users in the system
$\mathcal{U}_j(t)$	set of users allocated to BS s_j in time slot t
$\mathcal{U}^k_j(t)$	set of users allocated to BS s_j on subchannel c^k_j in time slot t
w_i	the amount of computing resource required to serve user u_i. w_i is a $\|\mathcal{R}\|$-dimensional vector
$\Theta^k_j(t)$	SIC decoding order of users on subchannel c^k_j in time slot t
r_{req}	minimum user data rate requirement
η_1, η_2	weights of allocation delay cost and interference cost in the whole system cost

Appendix B
Proof of Lemma 1

Proof.

$$
\begin{aligned}
&\Delta(D(t)) = \mathbb{E}\{L(D(t+1)) - L(D(t))|D(t)\}\\
&= \mathbb{E}\left\{\frac{1}{2}\sum_{u_i\in\mathcal{U}}(D_i(t) + r_{req} - r_i(t))^2 - \frac{1}{2}\sum_{u_i\in\mathcal{U}} D_i(t)^2|D(t)\right\}\\
&= \mathbb{E}\left\{\sum_{u_i\in\mathcal{U}}\Big(D_i(t)(r_{req} - r_i(t)) - r_{req}r_i(t)\right.\\
&\qquad\qquad\left.\frac{r^2_{req}}{2} + \frac{r^2_i(t)}{2}\Big)|D(t)\right\}\\
&\leq C + \mathbb{E}\left\{\sum_{u_i\in\mathcal{U}}\Big(D_i(t)(r_{req} - r_i(t)) - r_{req}r_i(t) + \right.\\
&\qquad\qquad\left.\frac{r^2_i(t)}{2}\Big)|D(t)\right\}
\end{aligned}
\tag{17}
$$

where $O = \frac{r^2_{req}}{2}$ is a constant. By adding $V\big(\eta_1 M_i(\mathbf{a}(t)) + \eta_2 I_i(\mathbf{a}(t), \mathbf{p}(t))\big)$ to both sides of (17), we complete the proof. □

Appendix C
Proof of Theorem 1

Proof. We first introduce the definition of *C-addictive approximation* [31] of an algorithm that minimizes a drift-plus-penalty. The algorithm is the one that determines an allocation strategy in each time slot. That strategy should yield a conditional expected value on the right-side of (8) within a positive constant C from the infimum over all possible allocation strategies.

Let us assume that (10) holds. By taking expectations of (10) and applying the law of iterated expectations, we have:

$$
\begin{aligned}
&\mathbb{E}\{L(D(t+1))\} - \mathbb{E}\{L(D(t))\} - V\,\mathbb{E}\{y(t)\}\\
&\leq O + C - \epsilon\sum_{u_i\in\mathcal{U}}\mathbb{E}\{D_i(t)\} - Vy_{opt}
\end{aligned}
\tag{18}
$$

This inequality is valid for all time slots t. Summing it over $t \in \{0, 1, ..., T-1\}$ and employing the law of telescoping sums yield:

$$
\begin{aligned}
&\mathbb{E}\{L(D(T))\} - \mathbb{E}\{L(D(0))\} - V\sum_{t=0}^{T-1}\mathbb{E}\{y(t)\}\\
&\leq T(O + C) - \epsilon\sum_{t=0}^{T-1}\sum_{u_i\in\mathcal{U}}\mathbb{E}\{D_i(t)\} - VTy_{opt}
\end{aligned}
\tag{19}
$$

As $L(D(0)) = 0$, $L(D(T)) \geq 0$, and $y(t) \leq y_{max}$, $\forall t$, dividing both sides of the above inequality by ϵT and rearranging terms give us:

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\sum_{u_i\in\mathcal{U}}\mathbb{E}\{D_i(t)\} - \frac{\mathbb{E}\{L(D(0))\}}{\epsilon T}\\
&\qquad\qquad\leq \frac{O + C + V(y_{max} - y_{opt})}{\epsilon}
\end{aligned}
\tag{20}
$$

The proof of the accumulated data rate bound (12) completes by letting $T \to \infty$.

To prove the bound of the long-term system cost (11), we rearrange the terms of (19) and divide it by VT:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{y(t)\} \geq \frac{\mathbb{E}\{L(D(T))\} - \mathbb{E}\{L(D(0))\}}{VT} - \frac{O+C}{V} + \frac{\epsilon}{VT}\sum_{t=0}^{T-1}\sum_{u_i\in\mathcal{U}}\mathbb{E}\{D_i(t)\} + y_{opt} \tag{21}$$

As $L(D(0)) = 0$, $L(D(T)) \geq 0$, and $D_i(t) \geq 0, \forall t$, we have:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{y(t)\} \geq y_{opt} - \frac{O+C}{V} \tag{22}$$

The proof of the long-term system cost bound (11) completes by letting $T \to \infty$. □

Appendix D
Proof of Theorem 2

Proof. In each iteration of Algorithms 2 and 3, a user $u_i \in \mathcal{U}$ changes its current decision $\mathbf{a}_i(t)$ to a new decision $\mathbf{a}'_i(t)$ to decrease the cost $C(.)$, or the potential function $\phi(\mathbf{a}_i(t), \mathbf{p}_i(t))$. Intuitively, in order to find the convergence rate, one needs to find the maximum value of the cost $C(.)$ or potential function $\phi(.)$, then divide it by the minimum decrease in cost $C(.)$ that might occur when this user changes its decision in the next iteration.

First, we find the maximum value of the cost $C(.)$ or potential function $\phi(.)$. According to (**??**), we have:

$$0 \leq \phi(\mathbf{a}_i(t), \mathbf{p}_i(t)) \leq |\mathcal{U}|(\eta_1 M_{max} + \eta_2 I_{max})$$

because the term $\sum_{u_i\in\mathcal{U}}(\eta_1 M_{max} + \eta_2 I_{max})\mathbb{1}_{\mathbf{a}_i=(0,0)}$ in the potential function $\phi(\mathbf{a}_i(t), \mathbf{p}_i(t))$ is always greater than the other terms combined.

Next, we need to find the minimum decrease in cost $C(.)$ that might occur when this user changes its decision in the next iteration of Algorithms 2. The minimum cost decrease happens when a user's new decision meets all the conditions in either one of the following cases:

Case 1:

- The user wishes to switch to another edge server/BS only to lower its allocation delay cost. The minimum improvement of the allocation delay occurs when the user currently has to wait for one time slot to start using the application in the current edge server, and the new edge server is ready to serve the user straight away without any delay.
- There are no changes in the data rates of any existing user. And the switch does not affect any other user.

Case 2:

- The user wishes to switch to another subchannel only to lower its intra-cell interference (and consequently improve its data rate). The minimum improvement of intra-cell interference occurs when the new subchannel has at least one user fewer than the current subchannel. The intra-cell interference incurred by this one user is at least $|h_{min}|^2 p$, where $|h_{min}|^2$ is the lowest possible channel gain of a user.
- This user does not suffer from inter-cell interference.
- The user is already allocated to a BS with an empty queue and does not wish to move to another BS. Thus, the allocation delay cost remains unchanged at zero when $\mathbf{a}_i(t)$ is updated to $\mathbf{a}'_i(t)$. And the switch does not affect any other user.

In Algorithm 2, users' transmit power is fixed at p. We let $\Delta C(.) = C'(.) - C''(.)$ be the **minimum** cost decrease between an iteration ($C'(.)$) and the consequent iteration ($C''(.)$). In Case 1, $\Delta C(.) = V\eta_1 M'_i(\mathbf{a}(t)) - V\eta_1 M''_i(\mathbf{a}(t)) = V\eta_1 1 - V\eta_1 0 = V\eta_1$. In Case 2, we have:

$$\Delta C(.) = \frac{r_i'^2(t) - r_i''^2(t)}{2} + D_i(t)(r''_i(t) - r'_i(t)) + r_{req}(r''_i(t) - r'_i(t)) + V\eta_2|h_{min}|^2 p$$

Since $r''_i(t) > r'_i(t)$ and $D_i \geq 0, \forall u_i, \forall t$, the minimum value of $D_i(t)(r''_i(t) - r'_i(t))$ is 0. For any user u_i, the minimum value of $r'_i(t)$ is $(B/K)\log_2\big(1 + \frac{|h_{min}|^2 p}{|h_{min}|^2 p + \sigma^2}\big)$; and the minimum value of $r''_i(t)$ is $(B/K)\log_2\big(1 + \frac{|h_{min}|^2 p}{\sigma^2}\big)$. Combining Case 1 and Case 2, we have $\Delta C(.) = min\{V\eta_1, Z\}$, where $Z = \frac{r_i'^2(t) - r_i''^2(t)}{2} + r_{req}(r''_i(t) - r'_i(t)) + V\eta_2|h_{min}|^2 p$ calculated as aforementioned. Algorithm 2 will terminate by driving the potential function to a minimum point (Nash equilibrium) within at most $\frac{|\mathcal{U}|\big(\eta_1 M_{max} + \eta_2 I_{max}\big)}{min\{V\eta_1, Z\}}$ iterations.

Finding the convergence rate of Algorithm 3 follows the same process above. In each iteration of Algorithm 3, a user $u_i \in \mathcal{U}$ changes its current decision $\mathbf{p}_i(t)$ to a new decision $\mathbf{p}'_i(t)$ to decrease the cost $C(.)$. We need to find the minimum decrease in cost $C(.)$ that might occur when this user changes its decision. Adjusting power level has no influence on the allocation delay and interference of that user so we have:

$$\begin{aligned}\Delta C(.) &= \frac{r_i'^2(t) - r_i''^2(t)}{2} + D_i(t)(r''_i(t) - r'_i(t)) + r_{req}(r''_i(t) - r'_i(t))\\ &= \frac{r_i'^2(t) - r_i''^2(t)}{2} + (D_i(t) + r_{req})(r''_i(t) - r'_i(t))\end{aligned}$$

Since $D_i(t) + r_{req} \geq 0$, we have $\Delta C(.) = \frac{r_i'^2(t) - r_i''^2(t)}{2}$. For any user u_i, the minimum value of $r'_i(t)$ is $(B/K)\log_2\big(1 + \frac{|h_{min}|^2 l_{min}}{\sigma^2}\big)$; and the minimum value of $r''_i(t)$ is $(B/K)\log_2\big(1 + \frac{|h_{min}|^2 l_{min+1}}{\sigma^2}\big)$, where l_{min} and l_{min+1} are the lowest and second lower power levels in $\mathcal{L}$, respectively. For the brevity of notation, we use X to denote the value of $\Delta C(.)$ in Algorithm 3. We can see that the minimum cost decrease happens when this user increases its transmit power by one level from the lowest level l_{min}. Similar to Algorithm 2, Algorithm 3 will terminate by driving the potential function to a minimum point (Nash equilibrium) within at most $\frac{|\mathcal{U}|\big(\eta_1 M_{max} + \eta_2 I_{max}\big)}{X}$ iterations, where X is calculated as aforementioned. □

Appendix E
Proof of Theorem 3

Proof. **Case 1:** In a time slot, the system cost incurred by any arbitrary allocation strategy $(\mathbf{a}(t), \mathbf{p}(t))$ is clearly always higher than the system cost incurred by an optimal allocation strategy, i.e., $Y(\mathbf{a}(t),\mathbf{p}(t)) \geq Y(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t))$, hence PoA ≥ 1.

Case 2: Next, we analyze the bounds on the system cost of an arbitrary Nash equilibrium and an optimal solution in a time slot. First, we find the maximum interference and allocation delay that a user can experience on a subchannel in a BS/edge server, which can be derived by analyzing the worst-case scenario. In the worst-case scenario, the interference received by any user u_i is at most $|h_{max}|^2|\mathcal{U}|l_{max}$, where $|h_{max}|^2$ is the highest possible channel gain of a user, and $l_{max} \in \mathcal{L}$ is the highest power level of a user. The allocation delay experienced by a user u_i is at most $\frac{[|\mathcal{U}|-N_{min}]_+}{N_{min}j/\ell}$, where N_{min} is the minimum edge server service rate (which belongs to the most resource-limited edge server). Therefore, for an arbitrary Nash equilibrium $(\mathbf{a}(t),\mathbf{p}(t))$, its system cost $Y(\mathbf{a}(t),\mathbf{p}(t))$ always satisfies:

$$\begin{aligned} Y(\mathbf{a}(t),\mathbf{p}(t)) \leq & \max_{\mathbf{a}(t)\in\mathcal{A}(t),\mathbf{p}(t)\in\mathcal{P}(t)} \sum_{u_i\in\mathcal{U}} \big(\eta_1 M_i(\mathbf{a}(t)) \\ & \qquad\qquad + \eta_2 I_i(\mathbf{a}(t),\mathbf{p}(t))\big) \\ \leq & |\mathcal{U}|\left(\eta_1 \frac{[|\mathcal{U}|-N_{min}]_+}{N_{min}/\ell} + \eta_2(|h_{max}|^2|\mathcal{U}|l_{max}+\sigma^2)\right) \end{aligned} \tag{23}$$

For an optimal solution $(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t))$, its cost $Y(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t))$ always satisfies:

$$\begin{aligned} Y(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t)) = & \sum_{u_i\in\mathcal{U}} \big(\eta_1 M_i(\mathbf{a}^{opt}(t)) + \eta_2 I_i(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t))\big) \\ & \overset{\ddagger}{\geq} |\mathcal{U}|\eta_2\sigma^2 \end{aligned} \tag{24}$$

where the inequality ‡ happens because $M_i(\mathbf{a}^{opt}(t)) \geq 0$ and $I_i(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t)) \geq \sigma^2, \forall u_i \in \mathcal{U}$.

Since $Y(\mathbf{a}(t),\mathbf{p}(t)) \geq Y(\mathbf{a}^{opt}(t),\mathbf{p}^{opt}(t)) \geq 0$, combined with (23) and (24), we have:

$$\text{PoA} \leq \frac{\eta_1 \frac{[|\mathcal{U}|-N_{min}]_+}{N_{min}/\ell} + \eta_2(|h_{max}|^2|\mathcal{U}|l_{max}+\sigma^2)}{\eta_2\sigma^2}$$

The combination of Case 1 and Case 2 completes the proof.

□