to appear in IEEE Transactions on Parallel and Distributed Systems (c) 2021 IEEE

Data, User and Power Allocations for Caching in Multi-Access Edge Computing

Xiaoyu Xia, Feifei Chen, Qiang He, *Senior Member, IEEE*, Guangming Cui, John Grundy, *Senior Member, IEEE*, Mohamed Abdelrazek, Xiaolong Xu, and Hai Jin, *Fellow, IEEE*

Abstract—In the multi-access edge computing (MEC) environment, app vendors' data can be cached on edge servers to ensure low-latency data retrieval. Massive users can simultaneously access edge servers with high data rates through flexible allocations of transmit power. The ability to manage networking resources offers unique opportunities to app vendors but also raises unprecedented challenges. To ensure fast data retrieval for users in the MEC environment, edge data caching must take into account the allocations of data, users, and transmit power jointly. We make the first attempt to study the Data, User, and Power Allocation (DUPA^3) problem, aiming to serve the most users and maximize their overall data rate. First, we formulate the DUPA^3 problem and prove its $\mathcal{NP}$-completeness. Then, we model the DUPA^3 problem as a potential DUPA^3 game admitting at least one Nash equilibrium and propose a two-phase game-theoretic decentralized algorithm named DUPA^{3}Game to achieve the Nash equilibrium as the solution to the DUPA^3 problem. To evaluate DUPA^{3}Game, we analyze its theoretical performance and conduct extensive experiments to demonstrate its effectiveness and efficiency.

Index Terms—edge computing, data allocation, user allocation, power allocation, optimization, multi-access

1 Introduction

Due to the exponential growth of mobile and Internet-of-Things (IoT) devices like smart phones and smart vehicles over the last decade, multi-access edge computing (MEC) is emerging as the novel distributed computing paradigm to tackle the unprecedented challenges raised by the enormous network traffic. In the MEC environment, an edge server powered by a cluster of physical machines is attached to each base station geographically close to users [1]. In this way, mobile and IoT application vendors (*app vendors*) can hire storage and computing capacities on edge servers for hosting their apps (*edge apps*) to serve nearby users with low latency [2].

When users access edge apps, a large volume of mobile data is transmitted via edge servers between the cloud and users' devices. Caching app data, especially popular ones like popular VR videos from Facebook Horizon[1], can considerably reduce the delay in users' data retrieval [3]. Moreover, the transferred data from the cloud to users can also be significantly reduced by caching app data on edge servers [4]. This way, the data transmission costs are lowered [5]. In the MEC environment, the new challenges of *edge data caching* (EDC) are starting to attract researchers' attention in recent years, who aim to maximize caching performance and/or minimize caching cost in general [2], [4].

Given a group of users and their data requests in a specific area, a straightforward solution for the app vendor is to cache all the requested data on each edge server. However, due to the size limits, edge servers usually have limited storage capacities [2], [6]. The competition among app vendors makes it often impossible to cache all the requested data on individual edge servers. In this case, reserving cache spaces on individual edge servers is a common practice [7]. Then, popular app data can be cached for users to retrieve who are allocated appropriate transmit power to ensure their data rates. Unallocated users that have to access the remote cloud will suffer from high latency and incur user attrition costs to app vendors [3].

In the MEC environment, networking resources play a critical role in impacting users' achievable data rates when users are retrieving cached data from edge servers. Networking resource management has been widely studied in research that combines cloud computing and radio access network [8], [9]. Very recently, researchers are starting to investigate new challenges in the MEC environment under multiple access schemes with consideration of networking resources, e.g., data offloading [10], and computation offloading [11]. However, existing EDC studies have predominantly focused on storage resources, and ignored or over-simplified networking resources during app data retrieval in the MEC environment where multiple access schemes are enabled to power the 5G wireless network.

The Non-Orthogonal Multiple Access (NOMA) scheme is a new multiple access scheme for 5G. It improves spectral efficiency significantly and provides connectivity for massive users by allowing non-zero cross-correlation signals,

- X. Xia, F. Chen and M. Abdelrazek are with School of Information Technology, Deakin University, Melbourne, Australia. E-mail: xiaoyu.xia@deakin.edu.au; feifei.chen@deakin.edu.au; mohamed.abdelrazek@deakin.edu.au.
- Q. He and G. Cui are with School of Software and Electrical Engineering, Melbourne, Swinburne University of Technology, Australia. E-mail: qhe@swin.edu.au; gcui@swin.edu.au.
- J. Grundy is with Faculty of Information Technology, Monash University, Australia. E-mail: john.grundy@monash.edu.
- X. Xu is with School of Computer and Software, Nanjing University of Information Science and Technology, China. Email: njuxlxu@gmail.com.
- H. Jin is with School of Computer Science and Technology, Huazhong University of Science and Technology, China. Email: hjin@hust.edu.cn.

[1] https://www.oculus.com/facebookhorizon/

compared with conventional orthogonal multiple access schemes [12]. Under the NOMA scheme, multiple channels are available on each base station (edge server). Each channel can accommodate multiple users simultaneously, whose data rates are guaranteed through appropriate transmit power allocation [13], [14], [15]. In a NOMA-based MEC environment, app vendors for the first time can manage the storage and networking resources jointly for ensuring fast app data retrieval, by submitting their strategies to the edge infrastructure provider for implementation [16]. This offers many new opportunities, and in the meantime introduces unprecedented challenges that app vendors have never encountered before MEC. As NOMA becomes widely acknowledged in both academia and industry, researchers are starting to investigate its impact on MEC problems, e.g., computation offloading [17] and user allocation [15].

The EDC problem has been investigated intensively from the edge infrastructure provider's perspective with various optimization objectives, e.g., minimum delay cost [18] or maximum data sharing efficiency [19]. However, app vendors, as key stakeholders in the NOMA-based MEC environment, must consider the allocation of their own data, their own users and the transmit power jointly and systematically when formulating their EDC strategies. This paper makes the first attempt to investigate this joint data, user and power allocations (DUPA^3) problem, aiming to 1) maximize user coverage (**EDC Objective #1**) and 2) maximize users' overall data rate (**EDC Objective #2**). Its key contributions are:

- We model and formulate the DUPA^3 problem for app vendors and prove its $\mathcal{NP}$-completeness.
- We model the DUPA^3 as a potential game, and prove that this DUPA^3 game can admit at least one Nash equilibrium.
- We propose a two-phase decentralized algorithm, namely DUPA^{3}Game, to achieve the Nash equilibrium in a DUPA^3 game, and evaluate its performance theoretically and experimentally.

The paper is structured as follows. We provide an example in Section 2 to motivate the research. The DUPA^3 problem is formulated in Section 3. In Section 4, we formulate the DUPA^3 game and present a two-phase decentralized algorithm to achieve the Nash equilibrium in a DUPA^3 game. In Section 5, we evaluate DUPA^{3}Game theoretically and experimentally. Section 6 reviews the related work and Section 7 concludes the paper.

2 Motivating Example

An example EDC scenario in the MEC environment is shown in Fig. 1, involving three edge servers, i.e., $\{s_1, s_2, s_3\}$, and nine users, i.e., $\{u_1, \cdots, u_9\}$ that request four data, i.e., $\{d_1, \cdots, d_4\}$. In the MEC environment, a user in the intersecting coverage of nearby edge servers can only retrieve data from one of them (*server coverage constraint*) [1]. For example, users u_6, u_7 and u_8 can only access edge server s_1 while u_3 can access either s_1 or s_3. In addition, the data pieces to be cached must not exceed reserved cache spaces on individual edge server, referred to as the *server capacity constraint* [2]. Let us take Fig. 1 as an example, assuming that d_1, d_2 and d_3 are cached on s_1. User u_8, who is requesting app data d_4, must obtain d_4 from the remote cloud and suffer the high latency. This incurs a significant loss for app vendors. **Thus, the first EDC objective (EDC Objective #1) is to minimize such losses by maximizing the *user coverage*, i.e., the number of users retrieving app data from edge servers.**

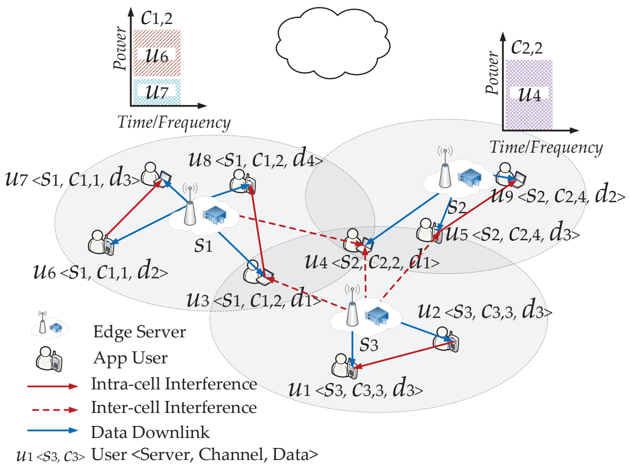

Power
$c_{1,2}$
u_6
u_7
Time/Frequency
Power
$c_{2,2}$
u_4
Time/Frequency
u_8 <s_1, $c_{1,2}$, d_4>
u_7 <s_1, $c_{1,1}$, d_3>
u_9 <s_2, $c_{2,4}$, d_2>
s_2
u_5 <s_2, $c_{2,4}$, d_3>
s_1
u_4 <s_2, $c_{2,2}$, d_1>
u_6 <s_1, $c_{1,1}$, d_2>
u_3 <s_1, $c_{1,2}$, d_1>
u_2 <s_3, $c_{3,3}$, d_3>
s_3
u_1 <s_3, $c_{3,3}$, d_3>
Edge Server
App User
Intra-cell Interference
Inter-cell Interference
Data Downlink
u_1 <s_3, c_3> User <Server, Channel, Data>

Fig. 1. An Edge Data Caching Scenario. The interference between the target users and the users of other apps hosted on s_1, s_2 and s_3 can be modelled as background noise and thus is omitted in the figure.

Under the NOMA scheme, EDC must also consider transmit power allocation to users under the server coverage constraint to ensure their data rates for data retrieval with the interference. To highlight the importance of networking resources in EDC, let us assume in Fig. 1 that all three edge servers have cached all the requested app data - this ensures full user coverage. Now let us consider an EDC strategy that allocates users $\{u_1, u_2\}$ to edge server s_3 on channel $c_{3,3}$, $\{u_6, u_7\}$ to s_1 on $c_{1,1}$, $\{u_3, u_8\}$ to s_1 on $c_{1,2}$, $\{u_4\}$ to s_2 on $c_{2,2}$ and $\{u_5, u_9\}$ to s_2 on $c_{2,4}$. Multiple users communicating with edge servers simultaneously may incur 1) intra-cell interference among users allocated to the same channel on the same edge server; and 2) inter-cell interference among users allocated to the same channel on different edge servers. Let us take a look at the users allocated to s_1 and s_2, i.e., $\{u_3, ..., u_9\}$. Users u_3 and u_8 are allocated to $c_{1,2}$ on s_1. There is intra-cell interference between their communication with s_1, indicated by the solid red line between them. The same applies to u_6 and u_7, u_3 and u_8, as well as u_5 and u_9. There is also inter-cell interference received by users in coverage areas intersect. Take u_3, u_4 and u_5 for example. User u_4 is in the overlapping area of all the three edge servers. Since u_4 is allocated to $c_{2,2}$ on s_2, it receives the inter-cell interference from both s_1 and s_3. In the meantime, u_3's data rate is impacted by the inter-cell interference from s_2 and u_5's data rate is impacted by that from s_3. Users' channel conditions are impacted by both intra-cell and inter-cell interference. Consequently, their data rates are impacted. According to users' channel conditions, the transmit power must be properly allocated to ensure users' data rates under the NOMA scheme. **Thus, the**

second EDC objective (EDC Objective #2) is to maximize their overall data rate by properly allocating transmit power to users.

In general, those with strong channel conditions, e.g., short distance from edge servers and less interference, are usually given less transmit power than those with poor channel conditions [12], [20]. Fig. 1 presents the power allocation decisions for u_4, u_6 and u_7 as an example. Suffering the same interference, u_6 is allocated more transmit power than u_7, because it is more distant from u_7 and thus has a poorer channel condition. Compared with u_6 and u_7, u_4 is given higher transmit power because it is the only user allocated to s_2 on $c_{2,2}$.

A real MEC environment may be very large with hundreds (or more) users, edge devices and edge servers. Thus any solution to this MEC allocation problem must also be able to scale and be computed in a reasonable time to be useful.

3 System Model

Below we formulate a model for our DUPA^3 problem. Table 1 summarizes the main notations. Given $\mathcal{M}$ users $\mathcal{U} = \{u_1, u_2, \cdots, u_{\mathcal{M}}\}$, $\mathcal{F}$ data $D = \{d_1, d_2, \cdots, d_{\mathcal{F}}\}$ and $\mathcal{N}$ edge servers $S = \{s_1, s_2, \cdots, s_{\mathcal{N}}\}$ in an area, the DUPA^3 problem needs to define an EDC strategy that includes: 1) a data allocation strategy that allocates D across S; 2) a user allocation strategy that allocates $\mathcal{U}$ to S on appropriate wireless communication channels; and 3) a power allocation strategy that allocates the transmit power of S to $\mathcal{U}$.

Definition 1 (Data Allocation Strategy). Let $\tau_{i,f} \in \{0,1\}$ indicate whether d_f $(1 \leq f \leq \mathcal{F})$ is cached on s_i $(1 \leq i \leq \mathcal{N})$, a data allocation strategy is represented by $\tau = \{\tau_{1,1}, \cdots \tau_{1,\mathcal{F}}, \cdots, \tau_{\mathcal{N},\mathcal{F}}\}$.

Let $C_i = \{c_{i,1}, c_{i,2}, \cdots c_{i,\mathcal{K}}\}$ denote the channels available on edge server s_i, $\mathcal{B}_i$ and p_i denote the bandwidth and transmit power of edge server s_i. The bandwidth and transmit power available on channel $c_{i,k}$ can be denoted by $\mathcal{B}_{i,k}$ and $p_{i,k}$.

Definition 2 (User Allocation Strategy). Let $\mathcal{X}^j_{i,k} \in \{0,1\}$ indicate whether user u_j $(1 \leq u \leq \mathcal{M})$ is allocated to channel $c_{i,k}$. The user allocation decision for u_j can be represented by $\mathcal{X}^j = \{\mathcal{X}^j_{1,1}, \cdots \mathcal{X}^j_{1,\mathcal{K}}, \cdots, \mathcal{X}^j_{\mathcal{N},\mathcal{K}}\}$. The user allocation strategy is constituted by all user allocation decisions, denoted by $\mathcal{X} = \{\mathcal{X}^0, \cdots, \mathcal{X}^{\mathcal{M}}\}$.

Let $\mathcal{U}_{i,k}(\mathcal{X})$ denote the users allocated to $c_{i,k}$ by $\mathcal{X}$.

Definition 3 (Power Allocation Strategy). Given a user u_j, the power allocation decision for u_j is denoted as $p^j = \{p^j_{1,1}, \cdots, p^j_{1,\mathcal{K}}, \cdots, p^j_{\mathcal{N},\mathcal{K}}\}$, where $p^j_{i,k}$ is the transmit power allocated to u_j $(u_j \in \mathcal{U}_{i,k}(\mathcal{X}))$. The power allocation strategy is constituted by all users' power allocation decisions, denoted by $\mathbf{p} = \{p^1, \cdots, p^{\mathcal{M}}\}$.

In the MEC environment, a device usually has a minimum power requirement and a power upper bound for wireless communication. Correspondingly, each user $u_j \in \mathcal{U}$ has a minimum power δ^j_{min} and a maximum power δ^j_{max}. Thus, for u_j to be allocated to channel $c_{i,k}$, its allocated transmit power $p^j_{i,k}$ must fulfil:

$$\mathcal{X}^j_{i,k} \cdot \delta^j_{min} \leq p^j_{i,k} \leq \mathcal{X}^j_{i,k} \cdot \delta^j_{max} \tag{1}$$

TABLE 1
Summary of Notations

Notation	Description
$\mathcal{B}_i$	bandwidth of s_i
$\mathcal{B}_{i,k}$	bandwidth of $c_{i,k}$
C_i	set of channels on s_i
$c_{i,k}$	channel k on s_i
$cost$	system cost incurred by failure to allocate a user
D	set of data
d_f	data f
$\mathcal{F}$	number of data
$g^j_{i,k}$	channel gain of u_j from $c_{i,k}$
$\mathcal{K}$	number of channel
$l_{i,j}$	distance between s_i and u_j
$\mathcal{M}$	number of users
$\mathcal{N}$	number of edge servers
$\mathcal{O}_{i,k}$	list of users allocated to $c_{i,k}$, re-ordered by channel conditions
$\mathbf{p}$	power allocation strategy
p_i	total power of s_i
$p_{i,k}$	total transmit power on $c_{i,k}$
p^j	power allocation decision of u_j
$p^j_{i,k}$	variable indicates the amount of power allocated to u_j from s_i
$\mathcal{R}^j_{i,k}$	data rate of u_j allocated to $c_{i,k}$
$\mathcal{S}$	set of edge servers
s_i	edge server i
$\mathcal{U}$	set of users
$\mathcal{U}_i$	set of users covered by s_i
$\mathcal{U}_{i,k}(\mathcal{X})$	set of users allocated to $c_{i,k}$ based on $\mathcal{X}$
u_j	user j
$\mathcal{X}$	user allocation strategy
$\mathcal{X}^j$	user allocation decision of u_j
$\mathcal{X}^j_{i,k}$	binary variable indicating whether u_j is allocated to $c_{i,k}$
$\rho_{benefit}$	POA in terms of overall system benefit
ρ_{cost}	POA in terms of overall system cost
σ	joint user and power allocation strategy
σ_j	joint user and power allocation decision for u_j
σ_{-j}	joint user and power allocation decisions for all the users except u_j
α_i	available storage spaces on s_i
$\tau_{i,f}$	binary variable indicating whether d_f is cached on s_i
$\vartheta^j_{i,k}$	inter-cell interference of u_j allocated to $c_{i,k}$
$\gamma^k_{i,j}$	SINR for u_j allocated to $c_{i,k}$
ω	background noise variance
δ^j_{min}	minimum power constraint of u_j
δ^j_{max}	maximum power constraint of u_j

Moreover, the total allocated power on s_i must not exceed the available power p_i:

$$\sum_{c_{i,k} \in C_i} \sum_{u_j \in \mathcal{U}_{i,k}(\mathcal{X})} p^j_{i,k} \leq p_i \tag{2}$$

Cache spaces are reserved on edge servers to cache commonly requested app data requested by users of edge devices. Thus, the spaces occupied by cached data on s_i must not exceed the cache spaces reserved on s_i:

$$\sum_{d_f \in D} \tau_{i,f} \leq \alpha_i \tag{3}$$

3.1 System Cost Model

The **EDC Objective #1**, discussed in Section 2, is to maximize user coverage. This minimizes the overall *system cost*. According to the server coverage constraint discussed in Section 2, a user u_j can be allocated to an edge server s_i only when it is covered by s_i:

$$\mathcal{X}_{i,k}^{j} \leq \begin{cases} 1 & u_j \in \mathcal{U}_i \\ 0 & u_j \notin \mathcal{U}_i \end{cases}, 1 \leq k \leq \mathcal{K} \tag{4}$$

where $\mathcal{U}_i$ is the set of users covered by s_i. Take user u_3 in Fig. 1 as an example. It is covered by edge servers s_1 and s_3, but not by s_2. Accordingly, $\mathcal{X}_{1,k}^{3}$ and $\mathcal{X}_{3,k}^{3}$ can be 0 or 1 while $\mathcal{X}_{2,k}^{3}$ can only be 0.

Let $\varphi_{j,f} \in \{0, 1\}$ indicate whether user u_j requests data d_f. User u_j can be allocated to a channel of s_i only when its requested data d_f is cached on s_i:

$$\mathcal{X}_{i,k}^{j} \leq \sum_{d_f \in D} \varphi_{j,f} \cdot \tau_{i,f} \tag{5}$$

A user u_j can be allocated to one channel at most, there is:

$$\sum_{s_i \in \mathcal{S}} \sum_{c_{i,k} \in C_i} \mathcal{X}_{i,k}^{j} \leq 1 \tag{6}$$

Let σ_j denote the joint user and power allocation decision for u_j that combines $\mathcal{X}^j$ and p^j:

$$\begin{aligned} \sigma_j &= (\mathcal{X}^j, p^j) \\ &= \{(\mathcal{X}_{1,1}^{j}, p_{1,1}^{j}), \cdots (\mathcal{X}_{1,\mathcal{K}}^{j}, p_{1,\mathcal{K}}^{j}), \cdots, (\mathcal{X}_{\mathcal{N},\mathcal{K}}^{j}, p_{\mathcal{N},\mathcal{K}}^{j})\} \end{aligned} \tag{7}$$

In this way, the joint user and power allocation strategy for $\mathcal{U} = \{u_1, ..., u_\mathcal{M}\}$ can be represented by $\sigma = \{\sigma_1, \cdots, \sigma_\mathcal{M}\}$.

The failure to allocate a user incurs the system cost. Its value is to be determined domain-specifically based on the app vendor's priority for avoiding unallocated users. Let $cost$ denote the cost incurred by one unallocated user and $\mathcal{I}_{\{condition\}}$ denote the indicator function such that returns 0 if the $condition$ is false, otherwise 1. The system cost incurred by user u_j can be calculated with:

$$Z(\sigma_j) = \mathcal{I}_{\{\sigma_j = \sigma_0\}} \cdot cost \tag{8}$$

where $\sigma_0 = \{(0, 0), (0, 0), \cdots, (0, 0)\}$, indicating that a user is not allocated to any edge server.

3.2 System Benefit Model

As discussed in Section 2, **EDC Objective #2** is to maximize users' overall data rate. This maximizes the overall *system benefit*.

The NOMA scheme implements the successive interference cancellation (SIC) technique, where a user with worse channel condition treats the signals of users allocated to the same channel with better channel conditions as noise [13]. Let us assume that the users allocated to channel $c_{i,k}$, i.e., $\mathcal{U}_{i,k}(\mathcal{X})$, are ordered by their channel conditions from poor to strong, i.e. $\mathcal{O}_{i,k} = \{1, 2, \cdots, |\mathcal{U}_{i,k}(\mathcal{X})|\}$. Accordingly, the j-th user in $\mathcal{U}_{i,k}(\mathcal{X})$, where $n < j < m$, can decode u_n's signal, treating u_m's signal as noise. Let $g_{i,k}^{j}$ denote the channel gain between channel $c_{i,k}$ on edge server s_i and user u_j, capturing the impact of antenna gain, shadowing and path-loss [21]. This channel gain can be calculated with $l_{i,j}^{-loss} \lambda |\hat{h}_{i,k}^{j}|^2$, where $l_{i,j}$ is the distance between s_i and u_j, $loss$ is the path loss exponent, λ is the frequency dependent factor, $\hat{h}_{i,k}^{j} \sim \mathcal{CN}(0, 1)$ is the fading coefficient from u_j on $c_{i,k}$. In this way, the intra-cell interference received by u_j allocated to channel $c_{i,k}$ can be calculated with $g_{i,k}^{j} \sum_{t=j+1}^{|\mathcal{U}_{i,k}(\mathcal{X})|} p_{i,k}^{t}$ [22]. In addition, the inter-cell interference received by u_j allocated to channel $c_{i,k}$, denoted by $\vartheta_{i,k}^{j}$, is $\vartheta_{i,k}^{j} = \sum_{s_o \in \mathcal{S} \setminus s_i} g_{o,k}^{j} p_{o,k}$ [20].

In EDC scenarios, users try to retrieve app data, e.g., interactive VR/AR data, from edge servers. Therefore, we only focus on the downlink in this study. According to [15], [20], the downlink Signal-to-Interference-plus-Noise Ratio (SINR) for u_j's communication with s_i on channel $c_{i,k}$ is calculated with:

$$\gamma_{i,k}^{j} = \frac{g_{i,k}^{j} p_{i,k}^{j}}{\underbrace{g_{i,k}^{j} \sum_{t=j+1}^{|\mathcal{U}_{i,k}(\mathcal{X})|} p_{i,k}^{t}}_{\text{intra-interference}} + \underbrace{\vartheta_{i,k}^{j}}_{\text{inter-interference}} + \underbrace{\omega}_{\text{noise}}} \tag{9}$$

where ω is the variance of the additive white Gaussian noise.

Assume that user u_m is allocated to channel $c_{i,k}$. It has to decode all other users' signals, i.e. $\{u_j \in \mathcal{U} | p_{i,k}^{j} > p_{i,k}^{m}\}$, $\forall j, m \in \mathcal{O}_{i,k}, j < m$, because u_j's transmit power is higher than u_m's. SIC requires that u_m's data rate for decoding u_j's signal is not lower than u_j's [15], [20]:

$$\mathcal{R}_{i,k}^{m \to j} \geq \mathcal{R}_{i,k}^{j \to j} \tag{10}$$

where

$$\mathcal{R}_{i,k}^{m \to j} = \mathcal{B}_{i,k} log_2(1 + \frac{p_{i,k}^{j}}{\sum_{q=j+1}^{|\mathcal{U}_{i,k}(\mathcal{X})|} p_{i,k}^{q} + \frac{\vartheta_{i,k}^{m} + \omega}{g_{i,k}^{m}}}) \tag{11}$$

In Fig. 1, users u_3 and u_8 are allocated to channel $c_{1,2}$, i.e., the 2nd channel on edge server s_1. Since the transmit power allocated to u_3 is higher than that allocated to u_8, u_8's data rate for decoding u_3's signal is not lower than u_3's: $\mathcal{R}_{1,2}^{8 \to 3} \geq \mathcal{R}_{1,2}^{3 \to 3} = \mathcal{R}_{1,2}^{3}$.

Once (10) is fulfilled, SIC can be performed on user u_j to perfectly cancel the intra-cell interference received by u_j [23], [24]. Hence, u_j's available downlink data rate can be expressed by:

$$\begin{aligned} \mathcal{R}_{i,k}^{j} = \mathcal{R}_{i,k}^{j \to j} = \mathcal{B}_{i,k} \cdot log_2(1+ \\ \frac{p_{i,k}^{j}}{\sum_{q=j+1}^{|\mathcal{U}_{i,k}(\mathcal{X})|} p_{i,k}^{q} + \max\left\{ \frac{\vartheta_{i,k}^{m} + \omega}{g_{i,k}^{m}} \mid \forall m \geq j \right\}}) \end{aligned} \tag{12}$$

where $m, j \in \mathcal{O}_{i,k}$. This equation shows that u_j's data rate is equal to the users' minimum rates after u_j in $\mathcal{O}_{i,k}$ for decoding u_j's signal.

The joint user and power allocation decisions for all the users except u_j can be represented by $\sigma_{-j} = \{\sigma_1, \cdots, \sigma_{j-1}, \sigma_{j+1}, \cdots, \sigma_{\mathcal{M}}\}$. A user's data rate is determined by its and as well as other users' joint user and power allocation decisions. Given a data allocation strategy τ and other users' joint user and power allocation decisions σ_{-j}, the system benefit produced by allocating a specific user u_j is calculated with:

$$B_{\tau,\sigma_{-j}}(\sigma_j) = \mathcal{I}_{\{\sigma_j \neq \sigma_0\}} \sum_{d_f \in D} \varphi_{j,f} \cdot \sum_{s_i \in S} \sum_{c_{i,k} \in C_i} \mathcal{R}^j_{i,k} \tag{13}$$

where $\sigma_j = \{(0,0), \cdots, (\mathcal{X}^j_{i,k}, p^j_{i,k}), \cdots, (0,0)\}$ and $\mathcal{R}^j_{i,k}$ is calculated with (12).

3.3 Optimization Model

The DUPA^3 problem consists of finite variables with corresponding domains listing their possible values and constraints over those variables. Thus, it is a constrained optimization problem. A feasible strategy to a constrained optimization problem is to assign a set of values to all variables in its domain while satisfying all the constraints.

Given $\mathcal{M}$ users $\mathcal{U} = \{u_1, u_2, \cdots, u_{\mathcal{M}}\}$, $\mathcal{F}$ data $D = \{d_1, d_2, \cdots, d_{\mathcal{F}}\}$ and $\mathcal{N}$ edge servers $\mathcal{S} = \{s_1, s_2, \cdots, s_{\mathcal{N}}\}$ in a specific area, the DUPA^3 problem is formulated as below:

$$\textbf{EDC Objective \#1:} \quad \min \sum_{u_j \in \mathcal{U}} Z(\sigma_j) \tag{14}$$

$$\textbf{EDC Objective \#2:} \quad \max \sum_{u_j \in \mathcal{U}} B_{\tau,\sigma_{-j}}(\sigma_j) \tag{15}$$

$$\begin{aligned} s.t.: \quad & \tau_{i,f} \in \{0,1\}, \forall s_i \in S, d_f \in D \\ & \mathcal{X}^j_{i,k} \in \{0,1\}, \forall s_i \in S, c_{i,k} \in \bigcup_{s_i \in S} C_i, u_j \in \mathcal{U} \\ & 0 \leq p^j_{i,k} \leq p_i, \forall s_i \in S, c_{i,k} \in \bigcup_{s_i \in S} C_i, u_j \in \mathcal{U} \\ & (1), (2), (3), (4), (5), (6) \end{aligned}$$

In the MEC environment, there are many domain-specific ways to the trade-off between **EDC Objective #1** and **EDC Objective #2**. For example, an app vendor can allocate most of their users to edge servers without worrying whether their overall data rate is optimal. It can also allocate some users to the cloud to reduce the interference among the remaining users so that their overall data rate is maximized.

Now, we demonstrate the $\mathcal{NP}$-completeness of the DUPA^3 problem.

Theorem 1. The DUPA^3 problem is $\mathcal{NP}$-complete.

Proof The DUPA^3 problem is the generalization of the multiple knapsack (MK) problem [25]. In a classic MK problem, there are m items $U' = \{u'_1, \cdots, u'_m\}$ with benefit b_j for each item and n knapsacks $S' = \{s'_1, \cdots, s'_n\}$ with capacities w_i for each knapsack. The MK problem aims to maximize total benefit of selected items, while obeying that the maximum total weight of the chosen items must not exceed $\sum_{s'_i \in S'} w_i$. In the DUPA^3 problem, users can be regarded as items while all the channels on the edge servers can be regarded as knapsacks. As mentioned in Section 3.3, there are many ways to trade off between the objectives in the DUPA^3 problem. Usually, the two objectives can be sum to create a weighted combination as a new objective. This way, the new objective of the DUPA^3 problem can be regarded as the objective in the MK problem. The constraints of the DUPA^3 problem, including the cache space constraint and power resource limit, can be projected to the weights in the MK problem. This way, the DUPA^3 problem is reduced to a MK problem and it is $\mathcal{NP}$-complete. □

4 Game Formulation and Algorithm Design

We propose DUPA^{3}Game, a 2-phase game-theoretic algorithm in this section to solve the DUPA^3 problem effectively and efficiently. Game theory is adopted for design of DUPA^{3}Game for the following three main reasons.

- Game theory does not need centralized control from the remote cloud which inevitably incurs extra communication latency unacceptable in the MEC environment.
- The solution to a game can be sought in a decentralized manner because the decisions can be made for individual users in parallel. In this way, the burden of finding the central optimal solution can be lifted, and the EDC strategy can be formulated rapidly.
- Game theory has been proven to be a powerful tool for mitigating participants' multiple conflicting objectives.

Game theory has been widely employed to solve different problems in a variety of domains. The key is to design a suitable game-theoretical approach for a specific problem. Game-theoretical approaches tackling different problems can be profoundly different in their designs. In our study, we design DUPA^{3}Game to tackle the novel DUPA^3 problem specifically, taking into account the unique constraints of MEC, including the server coverage constraint and the server capacity constraint.

4.1 Game Formulation and Property

Similar to many studies based on game theory [1], [8], [15], [26], the DUPA^3 game simulate all users by the corresponding players to make decisions, following a specific benefit function to achieve **EDC Objective #1:** maximizing user coverage, and **EDC Objective #2**: minimizing users' overall data rate. Under the NOMA scheme, users with poor channel conditions need high transmit power to ensure their data rates. The allocation of such users to pursue Objective #1 may undermine other users' data rates profoundly and conflict with the pursuit of Objective #2. From Eq. 13, we can see that system benefit is measured based on allocated users' data rates. An unallocated user will not produce any system benefit, and in the meantime will incur the system cost as discussed in Section 3.1. Thus, DUPA^{3}Game is designed to pursue Objective #2 as it will also approach Objective #1 at the same time.

In DUPA^3 game, the players make decisions on channels and edge servers that corresponding users are allocated to, and how much transmit power they obtain, producing a joint user and power allocation decision $\sigma_j = (\mathcal{X}^j, p^j)$ for each $u_j \in \mathcal{U}$. As discussed in Section 3, an EDC strategy involves data, user and power allocation strategies. There are two phases in the DUPA^3 game. In Phase #1, the data allocation strategy τ is formulated by updating the joint user and power allocation strategy σ. For example, if user u_j is allocated to $c_{i,k}$, i.e., $\mathcal{X}^j_{i,k} \leftarrow 1$, data d_f requested by u_j will be cached on server s_i if it is not cached on s_i, i.e., $\tau_{i,f} = 1$. In Phase #2, the joint strategy σ is formulated by joint user and power allocation decisions. This way, τ and σ constitute the final EDC strategy.

Given σ_{-j}, a decision σ_j needs to be made for user u_j to achieve its maximum benefit in data rate calculated with benefit function (13):

$$\max B_{\tau,\sigma_{-j}}(\sigma_j) \tag{16}$$

The DUPA^3 problem is formulated as a game $\mathcal{G} = (S, \mathcal{U}, \{\mathcal{A}_j\}_{u_j \in \mathcal{U}}, \{B_{\tau,\sigma_{-j}}(\sigma_j)\}_{\sigma_j \in \mathcal{A}_j})$ based on (16), where $\mathcal{A}_j$ is u_j's finite set of possible joint user and power allocation decisions. The users might conflict with others in this game. If some users have been allocated to a specific edge server, others may be prevented from being allocated to it due to the limits on reserved cache spaces and inadequate transmit power. Take Fig. 1 as an example. Allocating all the users in s_1's coverage area, including u_3, u_4, u_6, u_7 and u_8, to the same channel on edge server s_1 may exhaust the transmit power on that channel and thus cannot ensure these users' data rates. To mitigate such conflicts, we need to ensure that at least one Nash equilibrium [27] is admitted by the DUPA^3 game:

Definition 4 (Nash Equilibrium). An strategy $\sigma^* = (\sigma^*_1, \sigma^*_2, \cdots, \sigma^*_{\mathcal{M}})$ is a Nash equilibrium if no decision can unilaterally be changed for increasing any individual user's benefit, i.e.,

$$B_{\tau^*,\sigma^*_{-j}}(\sigma^*_j) \geq B_{\tau^*,\sigma^*_{-j}}(\sigma_j), \forall \sigma_j \in \mathcal{A}_j, u_j \in \mathcal{U} \tag{17}$$

where τ^* is determined by σ^*.

It is important to ensure that the DUPA^3 game can admit one Nash equilibrium at least due to the following property [28]:

Property 1. The user allocation decision σ^*_j for u_j is the best choice in $\mathcal{A}_i$ based on σ_{-j}, if the strategy σ^* is a Nash equilibrium.

A Nash equilibrium can be applied as a self-enforcing strategy for the DUPA^3 game based on Property 1. Since the sticking agreements are in users' own self-interests, there is no need for a centralized enforcement [28]. We first introduce the potential game [29] here:

Definition 5 (Potential Game). In a potential game, there is a potential function $\pi(\sigma)$ fulfilling:

$$\begin{aligned} B_{\tau,\sigma_{-j}}(\sigma_j) &< B_{\tau,\sigma_{-j}}(\sigma'_j) \\ \Rightarrow \quad \pi(\sigma_j, \sigma_{-j}) &< \pi(\sigma'_j, \sigma_{-j}) \end{aligned} \tag{18}$$

for any $u_j \in \mathcal{U}$, $\sigma_j, \sigma'_j \in \mathcal{A}_j$ and $\sigma_{-j} \in \prod_{l \neq j} \mathcal{A}_l$.

The Nash equilibrium in a DUPA^3 game can be interpreted in another way. An EDC strategy σ^* is a Nash equilibrium if there is $B_{\tau^*,\sigma^*_{-j}}(\sigma^*_j) = \max_{\sigma_j \in \mathcal{A}_j} B_{\tau^*,\sigma^*_{-j}}(\sigma_j)$, $\forall u_j \in \mathcal{U}$. Therefore, in a potential game, the local optima to the potential function can also ensure at least one Nash equilibrium [29].

Now, we first prove Lemma 1 for demonstrating that the DUPA^3 game is a potential game.

Lemma 1. Given an EDC strategy $\sigma = \{\sigma_1, \cdots, \sigma_{\mathcal{N}}\}$, a user u_j can be allocated to channel $c_{i,k}$, if the interference received by u_j, calculated with $\mu^j_{i,k}(\sigma) \triangleq \sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g^j_{i,k} p^j_{i,k}$, is not higher than T_j, calculated with:

$$T_j = \frac{g^j_{i,k} p^j_{i,k}}{2^{\frac{\bar{\mathcal{R}}}{\mathcal{B}_{i,k}}} - 1} - \vartheta^j_{i,k} - \omega \tag{19}$$

where $\bar{\mathcal{R}} = \mathcal{B}_{i,k} \cdot log_2(\frac{\delta^j_{min}}{p_{i,k} - \delta^j_{min}})$.

We provide the proof of Lemma 1 in Appendix A.

We now prove that the DUPA^3 game is a potential game with Theorem 2 based on Lemma 1.

Theorem 2 (Potential DUPA^{3}Game). With the potential function $\pi(\sigma_j, \sigma_{-j})$ below, the DUPA^3 game is a potential game.

$$\begin{aligned} &\pi(\sigma_j, \sigma_{-j}) = -\frac{1}{2} \sum_{u_j \in \mathcal{U}} ((g^j_{i,k} p^j_{i,k} \cdot I_{\{\sigma_j \neq \sigma_0\}} + T_j \cdot I_{\{\sigma_j = \sigma_0\}}) \cdot \\ &(\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g^t_{i,k} p^t_{i,k} \cdot I_{\{\sigma_j \neq \sigma_0\}} + T_t \cdot I_{\{\sigma_j = \sigma_0\}})) \end{aligned} \tag{20}$$

We provide the proof of Theorem 2 in Appendix B.

4.2 Algorithm Design and Convergence Analysis

It is important for a potential game to achieve a Nash equilibrium with finite iterations [29]. Based on this, the DUPA^3 game employs a 2-phase process to find the Nash equilibrium that involves $\mathcal{M}$ users $\mathcal{U} = \{u_1, u_2, \cdots, u_{\mathcal{M}}\}$, $\mathcal{F}$ data $D = \{d_1, d_2, \cdots, d_{\mathcal{F}}\}$ and $\mathcal{N}$ edge servers $S = \{s_1, s_2, \cdots, s_{\mathcal{N}}\}$. The pseudo codes of Phase #1 and Phase #2 are presented in Algorithm 1 and Algorithm 2 respectively.

Phase #1: In this phase, DUPA^{3}Game employs Algorithm 1 to formulate the data allocation strategy by updating joint user and power allocation decisions for individual users $u_j, j = 1, ..., \mathcal{M}$ who are temporarily given the minimum power as required by δ^j_{min}. This way, DUPA^{3}Game can allocate the most users to edge servers to achieve **EDC Objective #1**. The algorithm initializes σ_j for each user and τ (Lines 1-4). Let $\tau(t)$ and $\sigma(t)$ denote τ and σ_j in the current iteration t. In each iteration, the algorithm first updates data allocation strategy $\tau(t)$ based on the current joint user and power allocation strategy $\sigma(t)$ (Line 6).

Next, for each user u_j, we calculate the current system benefit produced by σ_j (Lines 8). After that, Algorithm 1 attempts to find out all the possible joint user and power allocation decisions for u_j to be stored in $\mathcal{A}_j$ (created on Line 9). To do that, we use a loop (Lines 10-18) to inspect every edge server $si \in S$. Specifically, edge servers that do not cover u_j or do not have adequate cache spaces are excluded (Lines 11). Then, if any channels of the remaining edge servers have enough transmit power for u_j, the corresponding joint user and power allocation decisions are included into $\mathcal{A}_j$ (Lines 12-16). Among all the joint decisions in $\mathcal{A}_j$, the one $\sigma'_j \in \mathcal{A}_j$ that produces the highest benefit is

Algorithm 1 Phase #1 in DUPA^{3}Game Algorithm

```
1:  initialization
2:  set the joint user and power allocation decision set $\sigma_j = \{(0,0),\cdots(0,0)\}$ for each user $u_j$, and $\sigma = \{\sigma_1,\cdots,\sigma_{\mathcal{M}}\}$
3:  initialize data allocation decisions $\tau_{i,f} = 0$ for strategy $\tau = \{\tau_{1,1},\cdots\tau_{1,\mathcal{F}},\cdots,\tau_{\mathcal{N},\mathcal{F}}\}$
4:  end of initialization
5:  repeat
6:      update data allocation strategy $\tau(t)$ according to $\sigma(t)$
7:      for all $u_j \in \mathcal{U}$ do
8:          calculate current benefit $B_{\tau(t),\sigma_{-j}(t)}(\sigma_j)$
9:          create $\mathcal{A}_j \leftarrow \varnothing$
10:         for all $s_i \in S$ do
11:             if $u_j \in \mathcal{U}_i$ and ($u_j$'s requested data is cached on $s_i$ or adequate cache space on $s_i$) then
12:                 for all $c_{i,k} \in C_i$ do
13:                     if $\Delta p_{i,k} \geq \delta^j_{min}$ then
14:                         $\mathcal{A}_j = \mathcal{A}_j \bigcup\{\cdots,(\mathcal{X}^j_{i,k} = 1, p^j_{i,k} = \delta^j_{min}),\cdots\}$
15:                     end if
16:                 end for
17:             end if
18:         end for
19:         find the allocation decision $\sigma'_j \in \mathcal{A}_j$ that produces the highest benefit
20:         if $B_{\tau(t),\sigma_{-j}(t)}(\sigma'_j) > B_{\tau(t),\sigma_{-j}(t)}(\sigma_j)$ then
21:             send $\sigma'_j$ to contend and wait for the winner
22:             if $u_j$ wins then
23:                 update $\sigma_j$ with $\sigma'_j$
24:             end if
25:         end if
26:     end for
27: until no decision updates
28: return $\tau$ and $\sigma$
```

sent to contend for update if it produces a higher benefit than u_j's current joint user and power allocation decision.

In each iteration, one user's joint user and power allocation decision is randomly selected in a decentralized manner. The calculations for individual users in Lines 5-27 is performed in parallel. The iterations repeats until no any user requests to submit its decision. Data allocation strategy τ and the joint user and power allocation strategy σ are returned as the input of Algorithm 2, which aims to achieve **EDC Objective #2**.

In this phase, an implicit heuristic is employed to accelerate the convergence of the game. Briefly speaking, in each iteration of the game, each user will first try to find a nearby edge server that has already cached the data it requires and has adequate transmit power to ensure the user's minimum requirement. If such an edge server cannot be found, the user will try to find a nearby edge server with adequate caching spaces and transmit power for processing its data request. Based on this heuristic, the data allocation decisions are made in Phase #1. After that, the decision making in Phase #2 only needs to focus on user and power allocation.

Algorithm 2 Phase #2 in DUPA^{3}Game Algorithm

```
1:  receive $\tau$ and $\sigma = \{\sigma_1,\cdots,\sigma_{\mathcal{M}}\}$ from Algorithm 1.
2:  repeat
3:      for all $u_j \in \mathcal{U}$ do
4:          calculate current benefit $B_{\tau,\sigma_{-j}(t)}(\sigma_j)$
5:          create $\mathcal{A}_j \leftarrow \varnothing$
6:          for all $s_i \in S$ do
7:              if $u_j \in \mathcal{U}_i$ and $u_j$'s requested data is cached on $s_i$ then
8:                  for all $c_{i,k} \in C_i$ do
9:                      if $\Delta p_{i,k} \geq \delta^j_{min}$ then
10:                         $p^j_{i,k} = \min\{\delta^j_{max}, \Delta p_{i,k}\}$
11:                         $\mathcal{A}_j = \mathcal{A}_j \bigcup\{\cdots,(\mathcal{X}^j_{i,k} = 1, p^j_{i,k}),\cdots\}$
12:                     end if
13:                 end for
14:             end if
15:         end for
16:         find the allocation decision $\sigma'_j \in \mathcal{A}_j$ that produces the highest benefit
17:         if $B_{\tau,\sigma_{-j}(t)}(\sigma'_j) > B_{\tau,\sigma_{-j}(t)}(\sigma_j)$ then
18:             send $\sigma'_j$ to contend and wait for the winner
19:             if $u_j$ wins then
20:                 update $\sigma_j$ with $\sigma'_j$
21:             end if
22:         end if
23:     end for
24: until no decision updates
25: return $\tau$ and $\sigma$
```

Phase #2: In this phase we aim to achieve **EDC Objective #2**. The DUPA^{3}Game employs this algorithm for allocating more power to each individual user $u_j \in \mathcal{U}$ based on the data allocation strategy τ and joint user and power allocation strategy σ returned by Algorithm 1. Algorithm 2 starts with calculating the system benefit produced by the current user and power decision for u_j (Line 4). Then, it iterates through all the edge servers s_i to find all the channels that can accommodate u_j with higher transmit power than its minimum transmit power (Lines 6-15), and includes the corresponding joint user and power allocation decisions into $\mathcal{A}_j$ (created on Line 5).

Next, if the optimal joint user and power allocation decision in $\mathcal{A}_j$ produces higher system benefit than the current decision for u_j, it will be sent to contend for update (Lines 16-22). This phase completes when no more decision updates are needed for any users (Line 24). Finally, τ and σ are returned as the final EDC strategy for solving the DUPA^3 problem.

The DUPA^3 game should achieve a Nash equilibrium within finite iterations. Let $T_{max} \triangleq \max(T_j)$, $T_{min} \triangleq \min(T_j)$, $Q_j \triangleq g^j_{i,k}p^j_{i,k}$, $Q_{max} \triangleq \max(Q_j)$, $Q_{min} \triangleq \min(Q_j)$, ($i = 1,\cdots,\mathcal{N}$, $j = 1,\cdots,\mathcal{M}$ and $k = 1,\cdots,\mathcal{K}$). The upper bound of the total number of iterations, denoted by Y, can be quantified with Theorem 3.

***Theorem* 3.** The total number of iterations in DUPA^{3}Game is not more than $\frac{\mathcal{M}T^2_{max}}{2Q_{min}}$:

$$Y \leq \frac{\mathcal{M}T^2_{max}}{2Q_{min}}$$

We provide the proof of Theorem 3 in Appendix C.

Now, we can analyze the computational complexity of DUPA^{3}Game based on Theorem 3. The computational complexity of both iteration processes in Algorithm 1 and Algorithm 2 are $O(\mathcal{M}\mathcal{N})$. Since the maximum number of iterations is $\frac{\mathcal{M}T_{max}^2}{2Q_{min}}$, the computational complexity of DUPA^{3}Game is $O(\frac{\mathcal{M}^2\mathcal{N}T_{max}^2}{2Q_{min}})$. As defined above Theorem 3, T_{max} and Q_{min} are constants. In addition, $\mathcal{K}$ is the number of channels on individual edge servers. It is usually not a large number. Thus, the computational complexity of DUPA^{3}Game is $O(\mathcal{M}^2\mathcal{N})$.

5 Evaluation

In this section, we analyze the theoretical performance of DUPA^{3}Game first, and then we evaluate it against three representative approaches experimentally.

5.1 Theoretical Analysis

In every iteration of the DUPA^3 game, the allocation decisions for individual users are made in parallel. The decision to be updated in each iteration is determined through a random selection. Such non-deterministic selections possibly lead to more than one Nash equilibrium. Therefore, the performance of DUPA^{3}Game is based on its Price of Anarchy (POA), measured by the ratio of the central optimal EDC strategy over the worst Nash equilibrium's utility [30]. In the DUPA^3 game, we measure the utility by the overall system cost incurred and the overall system benefit produced, both of which are calculated based on the number of users allocated. Thus, we prove Lemma 2 first.

Lemma 2 (Number of Allocated User). For any Nash equilibria σ in the DUPA^3, the number of users allocated $num(\sigma)$ fulfills:

$$\lfloor T_{min}/Q_{max}\rfloor \leq num(\sigma) \leq \lfloor T_{max}/Q_{min}\rfloor + 1 \qquad (21)$$

We provide the proof of Lemma 2 in Appendix D.

5.1.1 POA in Overall System Cost

Let $\sigma^* = (\sigma_1^*, \sigma_2^*, \cdots, \sigma_{\mathcal{M}}^*)$ denote the central optimal EDC strategy, and $\mathcal{G}$ denote EDC strategies achieving various Nash equilibria. Now, we analyze the overall system cost of DUPA^{3}Game with Theorem 4 based on Lemma 2.

Theorem 4 (POA in Overall System Cost). Given the central optimal EDC strategy σ^* and an EDC strategy $\sigma \in \mathcal{G}$, the POA in overall system cost denoted by ρ_{cost} fulfills:

$$1 \leq \rho_{cost}(\sigma) \leq \frac{\mathcal{M} - \lfloor T_{min}/Q_{max}\rfloor}{\mathcal{M} - \lfloor T_{max}/Q_{min}\rfloor - 1} \qquad (22)$$

We provide the proof of Theorem 4 in Appendix E.

5.1.2 POA in Overall System Benefit

Another optimization objective in the DUPA^3 problem is to maximize the overall system benefit. Here, we prove Theorem 5 for analyzing the POA in overall system benefit of DUPA^{3}Game.

TABLE 2
Parameter Settings

	$\mathcal{N}$	$\mathcal{M}$	$\mathcal{F}$
Set #1	10, 20, $\cdots$, 50	200	10
Set #2	30	100, 150, $\cdots$, 300	10
Set #3	30	200	6, 8, $\cdots$, 14

Theorem 5 (POA in Overall System Benefit). Given the central optimal EDC strategy σ^* and an EDC strategy $\sigma \in \mathcal{G}$, the POA of DUPA^{3}Game in terms of overall system benefit, denoted by $\rho_{benefit}$, fulfills:

$$\frac{\mathcal{R}_{min}(\lfloor T_{min}/Q_{max}\rfloor)}{\mathcal{R}_{max}(\lfloor T_{max}/Q_{min}\rfloor + 1)} \leq \rho_{benefit}(\sigma) \leq 1 \qquad (23)$$

where $\mathcal{R}_{min}$ is the minimum data rate of allocated users by DUPA^{3}Game, and $\mathcal{R}_{max} = \max\{\mathcal{B}_{i,k}, \forall c_k \in C, s_i \in S\}$.

We provide the proof of Theorem 5 in Appendix F.

5.2 Experimental Evaluation

5.2.1 Competing Approaches

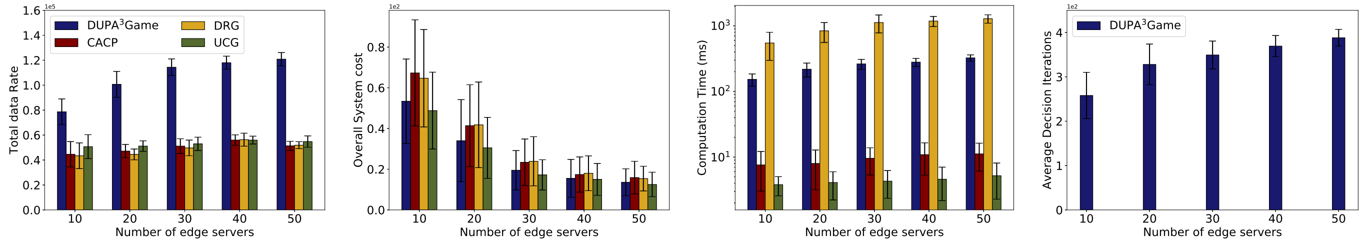
We compare the performance of DUPA^{3}Game against three representative approaches:

- *Centralized Algorithm for Cache Placement (CACP)* [31]: This greedy approach provides a near-optimal solution for minimizing the retrieval latency, considering the intra-cell interference and signal noise.
- *Data Rate Greedy (DRG)*: In each iteration of this approach, each user applies for its maximum power and sends its requests to all edge servers having adequate cache spaces. This approach always selects the data allocation decision that achieves the highest increase in total data rate until it satisfies constraint family (3). The user and power allocation decisions are determined after the data allocation decision is formulated.
- *User Coverage Greedy (UCG)*: In each iteration of this approach, the data allocation decision that allocates the most users is always selected until it satisfies constraint family (3). Once a data allocation decision is determined, UCG allocates the nearby users to selected edge server under the available power constraint. Similar to the DRG approach, each user applies for its maximum power.

5.2.2 Experiment Settings

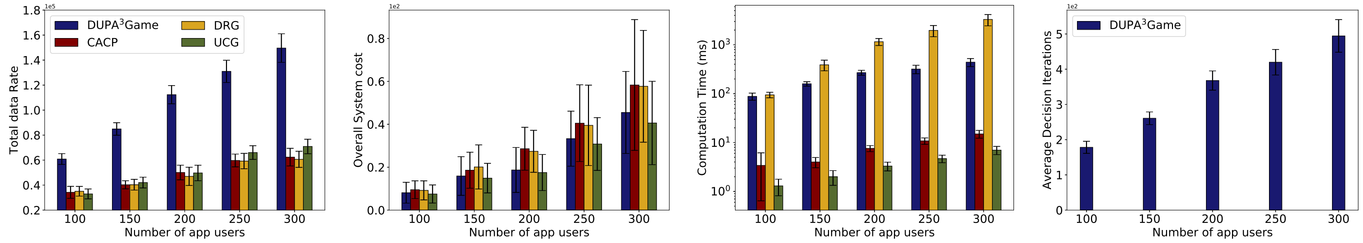
A real-world dataset, named EUA dataset[2], is used to conduct the experiments. This dataset contains the geographical locations of 130,000+ users and 90,000+ base stations in Australia. The experiments are conducted in the Melbourne CBD area with 816 users and 125 base stations. To simulate EDC scenarios generically, we use unitized data sizes. The maximum data cache storage is 20 units and each data is randomly sized from 1 to 4 units. As mentioned in Section

[2] https://github.com/swinedge/eua-dataset



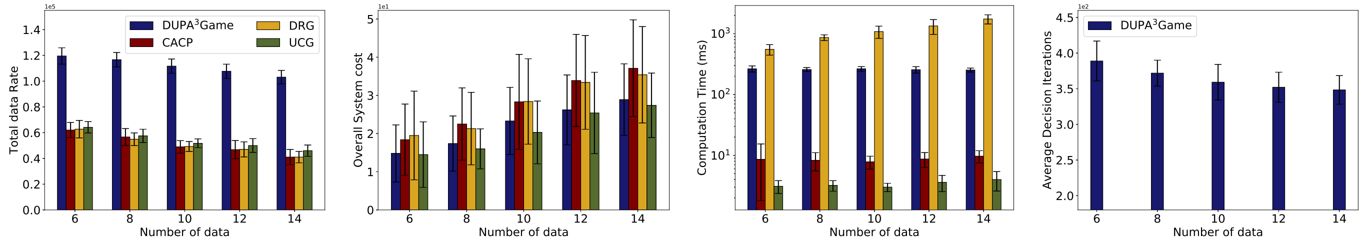
(a) $\mathcal{R}$ vs. $\mathcal{N}$ (b) $cost$ vs. $\mathcal{N}$ (c) $time$ vs. $\mathcal{N}$ (d) $iterations$ vs. $\mathcal{N}$

Fig. 2. Experiment Set #1



(a) $\mathcal{R}$ vs. $\mathcal{M}$ (b) $cost$ vs. $\mathcal{M}$ (c) $time$ vs. $\mathcal{M}$ (d) $iterations$ vs. $\mathcal{M}$

Fig. 3. Experiment Set #2



(a) $\mathcal{R}$ vs. $\mathcal{F}$ (Set #3) (b) $cost$ vs. $\mathcal{F}$ (Set #3) (c) $time$ vs. $\mathcal{F}$ (Set #3) (d) $iterations$ vs. $\mathcal{F}$ (Set #3)

Fig. 4. Experiment Set #3

3, each device has its own power range. Thus, in our experiments, the lower and upper bounds of each device's power, i.e., δ^j_{min} and δ^j_{max}, are randomly selected from 1 to 3 Watts and 3 to 5 Watts, respectively, similar to the device power settings employed in [13]. Following the same experiment setting in [8], [20], each server has 10 channels, each with channel bandwidth 1MHz, transmit power 100 Watts and background noise $\omega = -174dBm$, and we set $\lambda = 1$ and $loss = 3$ to calculate the channel gain $g^j_{i,k}$.

We simulate different EDC scenarios in the experiments by varying three parameters, summarized in Table 2: 1) the number of edge servers $\mathcal{N} = |S|$; 2) the number of users $\mathcal{M} = |\mathcal{U}|$; and 3) the number of data $\mathcal{F} = |D|$. In these sets, each experiment repeats 100 times when a setting parameter varies, and the average results are reported. The overall system cost and overall data rate are employed as the performance metrics for effectiveness evaluation, corresponding to **EDC Objective #1** and **EDC Objective #2** of the DUPA^3 problem.

Computation time is employed as the metric for efficiency evaluation. Please note that decisions in each iteration of the DUPA^3 game are simultaneously made. In this case, the time consumption of each iteration is calculated with the most time-consuming decision in that iteration. In addition, we also apply the number of decision iterations in DUPA^{3}Game as the convergence time to evaluate the efficiency of a game-theoretical approach [1], [8].

5.2.3 *Effectiveness*

Through comparison with CACP, DRG and UCG, Figures 2 - 4 demonstrate the effectiveness of DUPA^{3}Game. Overall, **DUPA^{3}Game achieves the highest overall data rate for users, while achieving the second lowest overall system cost**. Across three experiment sets, the average advantages of DUPA^{3}Game in the overall data rate are 116.63% over CACP, 119.37% over DRG and 104.57% over UCG. With respect to the overall system cost, its average performance loss compared with UCG, which achieves lowest overall system cost, is 8.82% in Set #1, 8.40% in Set #2 and 6.33% in Set #3.

Fig. 2 depicts the results of Set #1. DUPA^{3}Game achieves remarkably higher data rates than the other approaches in Fig. 2(a). When the number of edge servers increases, the overall data rates increases for all four approaches, from 78,642 to 120,848 by 53.67% for DUPA^{3}Game, from 44,624 to

51,304 by 14.97% for CACP, from 43,413 to 51,882 by 19.51% for DRG and from 50,699 to 54,802 by 8.93% for UCG. The reason is that, given more edge servers, users can be allocated to different channels to reduce the interference and receive higher transmit power. This is also the reason for the reduction in Fig. 2(b). Since UCG is a cost-oriented greedy approach that solely minimizes the overall system cost, it is not surprising that UCG achieves the lowest overall system cost in Fig. 2(b). DUPA^{3}Game achieves the second lowest overall system cost at 27.20 on average, compared with CACP's 33.08 and DRG's 32.76.

The experimental results with various numbers of users are shown in Fig. 3. The **overall data rate achieved by DUPA^{3}Game significantly and consistently outperforms those achieved by CACP, DRG and UCG** in Fig. 3(a). The performance of DUPA^{3}Game excels with significant advantages, 118.39% over CACP, 122.38% over DRG, 105.86% over UCG. In addition, with the increase in the number of users, the overall data rate achieved by DUPA^{3}Game increases much faster than the other approaches, i.e., 145.85% (DUPA^{3}Game) versus 82.14% (CACP), 72.91% (DRG) and 115.20% (UCG). This indicates that DUPA^{3}Game can utilize edge servers' resources effectively, including cache spaces and transmit power. When the number of users increases in Fig. 3(b), the overall system cost increases for all four approaches. The reason is that more users cannot be allocated due to the limited cache spaces and transmit power on edge servers in the experiments.

Fig. 4 demonstrates the experimental results with various numbers of data to cache. In Fig. 4(a), **DUPA^{3}Game achieves the highest overall data rate** again. With more data to be cached, the performance of all four approaches declines. Fig. 4(a) also shows that the performance of DUPA^{3}Game decreases much slower than other approaches, i.e. 13.75% (DUPA^{3}Game) versus 33.82% (CACP), 34.57% (DRG) and 28.27% (UCG). In Fig. 4(b), with the increasing number of data, the overall system costs achieved by all approaches increase, from 14.82 to 28.94 for DUPA^{3}Game, from 18.40 to 37.16 for CACP, from 19.58 to 35.42 for DRG and from 14.50 to 27.40 for UCG. Since the reserved cache spaces on edge servers are limited and do not suffice to cache all the data requested by nearby users, more requested data will result in more users retrieving data from the remote cloud.

5.2.4 Efficiency

Fig. 2(c), Fig. 3(c) and Fig. 4(c) illustrate the computation time taken by all the approaches in Sets #1, #2 and #3. In all the experiments, UCG takes the least time to complete, between 1ms and 7ms, and DRG takes the most time, between 95ms and 3,298ms. Compared with CACP and UCG and less than DRG, DUPA^{3}Game takes more time to complete, between 88ms and 440ms. This is the price it pays for the significant effectiveness advantages over the other approaches as shown and discussed in Section 5.2.3. In Fig. 2(c), when the number of edge servers $\mathcal{N}$ increases from 10 to 50 in Set #1, DUPA^{3}Game's computation time rises from 151ms to 325ms by 115.24% in Fig. 2(c). When the number of users $\mathcal{M}$ increases from 100 to 300 in Set #2, DUPA^{3}Game's computation time increases from 88ms to 441ms by 401.13% in Fig. 3(c). A comparison between Fig. 2(c) and Fig. 3(c) indicates that the impact of $\mathcal{M}$ is more significant than that of $\mathcal{N}$. These numerical results validate the analysis of DUPA^{3}Game's computational complexity in Section 4.2.

Fig. 2(d), Fig. 3(d) and Fig. 4(d) demonstrate the number of iterations in DUPA^{3}Game to achieve a Nash equilibrium. As shown in Fig. 2(d), **DUPA^{3}Game requires more iterations to converge with more edge servers**. The reason is that each user has more optional decision options. Thus, DUPA^{3}Game needs more iterations to move users around to reach a Nash equilibrium. Interestingly, DUPA^{3}Game takes fewer iterations with more data. Since the average cache spaces and user number are fixed, fewer users can be served by the data allocation strategy obtained by DUPA^{3}Game in Phase #1. Accordingly, fewer iterations are needed to finalize the decisions for all the users in Phase #2.

6 Related Work

Multi-access edge computing (MEC) allows app vendors deploying their data on edge servers to provide their users low latency service. MEC offers many unique advantages compared with cloud computing, however, it poses many new challenges for app vendors, e.g., edge data distribution [5], edge data integrity [32], collaborative edge computing [33], etc.

In recent years, researchers are starting to investigate edge data caching (EDC). Cao et al. [4] modeled the EDC problem as an auction between the edge infrastructure provider and users, where the former determined the allocation of cache spaces based on the latter's data evaluations. A method was proposed to calculate the cache space allocation and users' payments for cache data that optimized the data retrieval quality. Gharaibeh et al. [3] leveraged the ability of collaborative edge servers for minimizing app vendors' data caching cost. They proposed an online algorithm to determine how data should be retrieved and cached to fulfill users' data requests. Tran et al. [18] targeted edge video caching specifically and proposed two approaches, one for video data allocation and the other for video request scheduling. Their main objective was to optimize users' quality of experience by caching different bitrate versions of a video on edge servers. While edge data caching is a highly active research area where significant attentions are paid to users' quality of experience, none of the existing studies have taken into account the impacts of networking resources, which directly dictates users' data rates when retrieving data from edge servers.

Very recently, researchers are starting to realize the importance of networking resources in MEC. As the *de facto* radio access scheme for 5G, Non-Orthogonal Multiple Access (NOMA) promises low latency and massive connectivity among users and edge servers [12]. It complicates the allocation of users to base stations and has attracted a lot of researchers' attention [15], [20], [23], [24]. Fu et al. [23] studied a joint user allocation and power optimization problem under the NOMA scheme. They first proposed a distributed user selection and grouping approach to partition users into different groups. Then, they implemented a classic heuristic algorithm to optimize power consumption based on user allocation. Nguyen et al. [24] investigated a similar joint user allocation and power control problem under the NOMA scheme, aiming to maximize the spectral efficiency. They formulated it as a non-convex problem and proposed two heuristic algorithms to solve this problem.

In the NOMA-based MEC environment, app vendors must consider the allocation of data, users and transmit power jointly and systematically when formulating their EDC strategies. Without considering the characteristics of edge data caching, existing user allocation approaches designed under the NOMA scheme cannot be integrated into EDC approaches directly to tackle the new and challenging DUPA^3 problem. In this study, we proposed a novel two-phase game-theoretical approach named DUPA^{3}Game to solve this problem specifically.

7 Conclusion

In this paper, we have tackled the edge data caching problem in MEC environments. We formulated it as a joint data allocation, user allocation and power allocation (DUPA^3) problem and proved the $\mathcal{NP}$-completeness of this DUPA^3 problem. In order to be able to practically solve it, we proposed the novel DUPA^{3}Game, a two-phase game-theoretical approach that formulates a DUPA^3 game admitting a Nash equilibrium to solve the DUPA^3 problem. We analyzed the theoretical performance of DUPA^{3}Game and used a real-world dataset to evaluate DUPA^{3}Game experimentally. The results demonstrate the effectiveness and efficiency of DUPA^{3}Game for solving the DUPA^3 problem.

In the future work, we will also investigate the impact of users' dynamic participation in DUPA^3 scenarios where edge devices move around.

Acknowledgement

This research is partially funded by Australian Research Council Laureate Fellowship FL190100035 and Discovery Projects No. DP180100212, DP200102491.

References

[1] 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*, 2019.

[2] 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, 2020.

[3] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1863–1876, 2016.

[4] X. Cao, J. Zhang, and H. V. Poor, "An optimal auction mechanism for mobile edge caching," in *38th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 388–399.

[5] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, "Cost-effective app data distribution in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 31–44, 2021.

[6] 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*, pp. 1–1, 2020.

[7] X. Xia, F. Chen, J. Grundy, M. Abdelrazek, H. Jin, and Q. He, "Constrained app data caching over edge server graphs in edge computing environment," *IEEE Transactions on Services Computing*, 2021.

[8] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[9] Z. Ye, Y. Wang, S. He, C. Xu, and X.-H. Sun, "Sova: A software-defined autonomic framework for virtual network allocations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 116–130, 2020.

[10] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Transactions on Networking*, 2020.

[11] Z. Xu, L. Zhao, W. Liang, O. F. Rana, P. Zhou, Q. Xia, W. Xu, and G. Wu, "Energy-aware inference offloading for dnn-driven applications in mobile edge clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 4, pp. 799–814, 2020.

[12] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5g networks: Research challenges and future trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, 2017.

[13] S. Fu, F. Fang, L. Zhao, Z. Ding, and X. Jian, "Joint transmission scheduling and power allocation in non-orthogonal multiple access," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 8137–8150, 2019.

[14] X. Liu, Y. Liu, X. Wang, and H. Lin, "Highly efficient 3-d resource allocation techniques in 5g for noma-enabled massive mimo and relaying systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 12, pp. 2785–2797, 2017.

[15] 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.

[16] D. Sabella, V. Sukhomlinov, L. Trang, P. Paglierani, R. Rossbach, X. Li, Y. Fang, D. Druta, F. Giust, L. Cominardi, W. Featherstone, B. Pike, and S. Hadad, "Developing software for multi-access edge computing," *ETSI White Paper No. 20*, pp. 1–38, 2019.

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

[18] T. X. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 1965–1978, 2018.

[19] G. Luo, H. Zhou, N. Cheng, Q. Yuan, J. Li, F. Yang, and X. S. Shen, "Software defined cooperative data sharing in edge computing assisted 5g-vanet," *IEEE Transactions on Mobile Computing*, 2019.

[20] 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.

[21] 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.

[22] C. Singhal and S. De, *Resource allocation in next-generation broadband wireless access networks*. IGI Global, 2017.

[23] Y. Fu, M. Zhang, L. Salaün, C. W. Sung, and C. S. Chen, "Zero-forcing oriented power minimization for multi-cell miso-noma systems: A joint user grouping, beamforming, and power control perspective," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1925–1940, 2020.

[24] H. V. Nguyen, V.-D. Nguyen, O. A. Dobre, D. N. Nguyen, E. Dutkiewicz, and O.-S. Shin, "Joint power control and user association for noma-based full-duplex systems," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 8037–8055, 2019.

[25] C. Chekuri and S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 713–728, 2005.

[26] 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.

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

[28] C. A. Holt and A. E. Roth, "The nash equilibrium: A perspective," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 3999–4002, 2004.

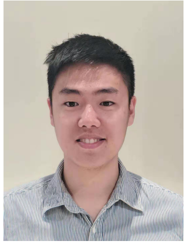
[29] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[30] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press Cambridge, 2005, vol. 174.

[31] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in fog-rans: From centralized to distributed algorithms," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7039–7051, 2017.

[32] 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*, 2020.

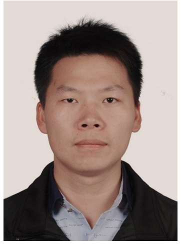
[33] L. Yuan, Q. He, S. Tan, B. Li, J. Yu, F. Chen, H. Jin, and Y. Yang, "Coopedge: A decentralized blockchain-based platform for cooperative edge computing," in *Proceedings of the Web Conference 2021*, 2021, pp. 2245–2257.



Xiaoyu Xia received his Master degree from The University of Melbourne, Australia in 2015. He is a PhD candidate at Deakin University. His research interests include edge computing, service computing, cloud computing, green computing and software engineering.

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.

Qiang He received his first PhD degree from Swinburne University of Technology, Australia, in 2009 and his second PhD degree in computer science and engineering from Huazhong University of Science and Technology, China, in 2010. He is an Associate Professor at Swinburne. His research interests include service computing, software engineering, cloud computing and edge computing. More details about his research can be found at https://sites.google.com/site/heqiang/.



Guangming Cui received his Master degree from Anhui University, China, in 2018. He is a PhD candidate at Swinburne University of Technology. His research interests include software engineering, edge computing and service computing.

John C. Grundy received the BSc (Hons), MSc, and PhD degrees in computer science from the University of Auckland, New Zealand. He is currently Australian Laureate Fellow and a professor of software engineering at Monash University, Melbourne, Australia. He is an associate editor of the IEEE Transactions on Software Engineering, the Automated Software Engineering Journal, and IEEE Software. His current interests include domain-specific visual languages, model-driven engineering, large-scale systems engineering, and software engineering education. More details about his research can be found at https://sites.google.com/site/johncgrundy/.

Mohamed Abdelrazek is an Associate Professor of Software Engineering and IoT at Deakin University. 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 https://sites.google.com/site/mohamedalmorsy/.

Xiaolong Xu received the Ph.D. degree in computer science and technology from Nanjing University, China, in 2016. He is currently an Associate Professor at Nanjing University of Information Science and Technology. His research interests include edge computing, Internet of Things (IoT), cloud computing, and big data.

Hai Jin is a Cheung Kung Scholars Chair Professor of computer science and engineering at Huazhong University of Science and Technology (HUST) in China. Jin received his PhD in computer engineering from HUST in 1994. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, peer-to-peer computing, network storage, and network security.

Appendix A
Proof of Lemma 1

Proof If user u_j can be allocated to the edge serve s_i on channel $c_{i,k}$, there is:

$$\mathcal{R}_{i,k}^{j} \geq \bar{\mathcal{R}}$$

Based on Eq. (11), this inequality can be converted to:

$$\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g_{i,k}^{j} p_{i,k}^{j} \leq \frac{g_{i,k}^{j} p_{i,k}^{j}}{2^{\frac{\bar{\mathcal{R}}}{\mathcal{B}_{i,k}}} - 1} - \vartheta_{i,k}^{j} - \omega = T_j$$

Thus, the interference received by user u_j is at most T_j. □

Appendix B
Proof of Theorem 2

Proof Assuming two allocation decisions σ_j and σ_j', that fulfill $B_{\tau,\sigma_{-j}}(\sigma_j) \leq B_{\tau,\sigma_{-j}(\sigma_j')}$, for user u_j. To prove this theorem, there are two cases according to (13): 1) $\sigma_j \neq \sigma_0$ and $\sigma_j' \neq \sigma_0$; and 2) $\sigma_j = \sigma_0$ and $\sigma_j' \neq \sigma_0$.

Case 1: $\sigma_j \neq \sigma_0$ and $\sigma_j' \neq \sigma_0$.

In this case, user u_j should be moved from channel $c_{i,k}$ to $c_{i',k'}$. Given $\mathcal{B}_{\sigma_{-j}}(\sigma_j) \leq \mathcal{B}_{\sigma_{-j}}(\sigma_j')$, based on Eq. (13), there is $\mathcal{R}_{i,k}^{j} < \mathcal{R}_{i',k'}^{j}$. According to Eq. (11), we can obtain:

$$\begin{aligned} &\mathcal{B}_{i,k} \log_2 \left(1 + \frac{p_{i,k}^{j}}{\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} p_{i,k}^{j} + \frac{\vartheta_{i,k}^{j} + \omega}{g_{i,k}^{j}}}\right) \\ &< \mathcal{B}_{i',k'} \log_2 \left(1 + \frac{p_{i',k'}^{j}}{\sum_{t=j+1}^{|\mathcal{U}_{i',k'}(\sigma)|} p_{i',k'}^{j} + \frac{\vartheta_{i',k'}^{j} + \omega}{g_{i',k'}^{j}}}\right) \end{aligned}$$

And we can obtain:

$$\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g_{i,k}^{t} p_{i,k}^{t} > \sum_{t=j+1}^{|\mathcal{U}_{i',k'}(\sigma)|} g_{i',k'}^{t} p_{i',k'}^{t}$$

Thus, the difference between the potentials produced by σ_j and σ_j' calculated with Eq. (20) can be represented as follows:

$$\begin{aligned} &\pi(\sigma_j, \sigma_{-j}) - \pi(\sigma_j', \sigma_{-j}) = \\ &g_{i,k}^{j} p_{i,k}^{j} \sum_{t=j+1}^{|\mathcal{U}_{i',k'}(\sigma)|} g_{i',k'}^{t} p_{i',k'}^{t} - g_{i,k}^{j} p_{i,k}^{j} \sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g_{i,k}^{t} p_{i,k}^{t} < 0 \end{aligned}$$

Case 2: $\sigma_j = \sigma_0$ and $\sigma_j' \neq \sigma_0$.

In this case, user u_j is unallocated and submits its update request for moving to channel $c_{i',k'}$. According to Lemma 1, we know that $\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g_{i,k}^{t} p_{i,k}^{t} \leq T_t$. Similar to Case 1, there is:

$$\begin{aligned} &\pi(\sigma_j, \sigma_{-j}) - \pi(\sigma_j', \sigma_{-j}) = \\ &g_{i',k'}^{j} p_{i',k'}^{j} \sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} g_{i',k'}^{t} p_{i',k'}^{t} - g_{i,k}^{j} p_{i,k}^{j} T_t < 0 \end{aligned}$$

Therefore, $\pi(\sigma_j, \sigma_{-j})$ is a potential function and the DUPA^3 game is a potential game. □

Appendix C
Proof of Theorem 3

Proof This proof is conducted for scenarios where T_j and Q_j are non-negative integers. The convergence time of DUPA^3 in scenarios where T_j and Q_j can be real numbers is evaluated through experiments in Section 5.2.

According to Eq. (20), there is:

$$-\frac{1}{2} \sum_{u_j \in \mathcal{U}} T_{max} \cdot T_{max} \leq \pi(\sigma_j, \sigma_{-j}) \leq 0$$

Since the total number of users is $\mathcal{M}$, we can obtain:

$$-\frac{1}{2} \mathcal{M} T_{max}^2 \leq \pi(\sigma_j, \sigma_{-j}) \leq 0 \tag{24}$$

If user u_j decides to update its allocation decision σ_j by σ_j', the corresponding benefit should be increased, i.e., $B_{\tau,\sigma_{-j}}(\sigma_j) < B_{\tau,\sigma_{-j}}(\sigma_j')$. This also leads to an increase in the potential with $\pi(\sigma_j, \sigma_{-j})$, denoted by ε_j, according to Definition 5:

$$\pi(\sigma_j', \sigma_{-j}) \geq \pi(\sigma_j, \sigma_{-j}) + \varepsilon_j \tag{25}$$

Now we try to prove $\varepsilon_j = Q_j$ for obtaining $\min_{u_j \in \mathcal{U}}(\varepsilon_j) = Q_{min}$, where $\min_{u_j \in \mathcal{U}}(\varepsilon_j)$ is the lowest value increased by updating a decision. Similar to Theorem 2, there are two cases when a decision is updated for a user: 1) $\sigma_j \neq \sigma_0$ and $\sigma_j' \neq \sigma_0$; and 2) $\sigma_j = \sigma_0$ and $\sigma_j' \neq \sigma_0$.

Case 1: $\sigma_j \neq \sigma_0$ and $\sigma_j' \neq \sigma_0$.

Based on Case 1 in the proof of Theorem 2, we can obtain:

$$\begin{aligned} &\pi(\sigma_j', \sigma_{-j}) - \pi(\sigma_j, \sigma_{-j}) \\ &= Q_j \cdot (\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} Q_t \cdot I_{\{\sigma_j \neq \sigma_0\}} - \sum_{t=j+1}^{|\mathcal{U}_{i',k'}(\sigma)|} Q_t' \cdot I_{\{\sigma_j' \neq \sigma_0\}}) > 0 \end{aligned} \tag{26}$$

Since $Q_j > 0$ is an integer for any $u_j \in \mathcal{U}$, there is:

$$\sum_{t=j+1}^{|\mathcal{U}_{i,k}(\sigma)|} Q_t \cdot I_{\{\sigma_j \neq \sigma_0\}} - \sum_{t=j+1}^{|\mathcal{U}_{i',k'}(\sigma)|} Q_t' \cdot I_{\{\sigma_j' \neq \sigma_0\}} \geq 1$$

Thus, according to Eq. (26), there is:

$$\pi(\sigma_j', \sigma_{-j}) \geq Q_j + \pi(\sigma_j, \sigma_{-j}) \geq Q_{min} + \pi(\sigma_j, \sigma_{-j})$$

Case 2: $\sigma_j = \sigma_0$ and $\sigma_j' \neq \sigma_0$.

Based on Case 2 in the proof of Theorem 2, we can obtain:

$$\begin{aligned} &\pi(\sigma_j', \sigma_{-j}) - \pi(\sigma_j, \sigma_{-j}) \\ =&Q_j \cdot (T_j - \sum_{t=j+1}^{|\mathcal{U}_{i',k'}(\sigma)|} Q_t \cdot I_{\{\sigma_j' \neq \sigma_0\}}) > 0 \end{aligned}$$

Similar to Case 1 in this proof, we can also obtain the following inequality:

$$\pi(\sigma_j', \sigma_{-j}) \geq Q_j + \pi(\sigma_j, \sigma_{-j}) \geq Q_{min} + \pi(\sigma_j, \sigma_{-j})$$

Therefore, according to (24) and (25), there is:

$$Y \leq \frac{\mathcal{M} T_{max}^2}{2 Q_{min}}$$

This indicates that the maximum convergence time of DUPA^{3}Game is at most $\frac{\mathcal{M} T_{max}^2}{2 Q_{min}}$. Therefore, Theorem 3 holds. □

Appendix D
Proof of Lemma 2

Proof As T_j is the maximum interference received by user u_j, Eq. (19) also holds for the optimal strategy σ^*:

$$T_j > \sum_{u_t \in \mathcal{U} \setminus \{u_j\}: \sigma_t^* = \sigma_j^*} Q_t$$

It follows $Q_j \geq Q_{min}$. Thus, for edge server s_i, there is:

$$(num_{s_i}(\sigma^*) - 1) \cdot Q_{min} \leq \sum_{u_t \in \mathcal{U} \setminus \{u_j\}: \sigma_t^* = \sigma_j^*} Q_t \leq T_j \leq T_{max}$$

This inequality can be converted to:

$$num_{s_i}(\sigma^*) \leq \lfloor T_{max}/Q_{min} \rfloor + 1$$

That is,

$$\sum_{u_j \in \mathcal{U}} I_{\{\sigma_j^* > 0\}} \leq \mathcal{M} \cdot (\lfloor T_{max}/Q_{min} \rfloor + 1)$$

For any feasible strategy $\sigma \in \mathcal{G}$ except σ^*, the total number of allocated users, calculated with $\sum_{s_i \in S} num_{s_i}(\sigma)$, is less than $\mathcal{M}$. Given a user $u_j \in \mathcal{U}$, it can be found that

$$T_j \leq \sum_{u_t \in \mathcal{U} \setminus \{u_j\}: \sigma_t = \sigma_j} Q_t$$

i.e.,

$$num_{s_i}(\sigma) \cdot Q_{max} \geq \sum_{u_t \in \mathcal{U} \setminus \{u_j\}: \sigma_t = \sigma_j} Q_l \geq T_j \geq T_{min}$$

Thus, there is $num_{s_i}(\sigma) \geq \lfloor T_{min}/Q_{max} \rfloor$ and $\sum_{u_j \in \mathcal{U}} I_{\{\sigma_j > 0\}} \geq \mathcal{M} \cdot \lfloor T_{min}/Q_{max} \rfloor$. Finally, we obtain the upper and lower bounds of the allocated users by strategy σ:

$$\lfloor T_{min}/Q_{max} \rfloor \leq num(\sigma) \leq \lfloor T_{max}/Q_{min} \rfloor + 1$$

Therefore, Lemma 2 is proved. □

Appendix E
Proof of Theorem 4

Proof Here, we calculate the POA of DUPA^{3}Game in system cost based on (14) and (21):

$$\begin{aligned} \rho_{cost} &\leq \frac{\mathcal{M} - \sum_{s_i \in S} num_{s_i}(\sigma)}{\mathcal{M} - \sum_{s_i \in S} num_{s_i}(\sigma^*)} \\ &\leq \frac{\mathcal{M} - \lfloor T_{min}/Q_{max} \rfloor}{\mathcal{M} - \lfloor T_{max}/Q_{min} \rfloor - 1} \end{aligned}$$

Since the overall system cost achieved by DUPA^{3}Game is not lower than that achieved by the optimal EDC strategy σ^*, ρ_{cost} is not lower than 1. Thus, Theorem 4 holds. □

Appendix F
Proof of Theorem 5

Proof Let us denote $\mathcal{R}_{min}$ as the minimum data rate of the users allocated by DUPA^{3}Game:

$$\mathcal{R}_{min} \geq \min\{\mathcal{B}_{i,k} \cdot log_2(\frac{p_{i,k}}{p_{i,k} - \delta_{min}^j}), \forall c_{i,k} \in C, s_i \in S\}$$

Given $\mathcal{R}_{min}$ and $\mathcal{R}_{max}$, the POA of DUPA^{3}Game in terms of the overall system benefit can be calculated with:

$$\begin{aligned} \rho_{benefit} &= \frac{\sum_{u_j \in \mathcal{U}} B_{\tau, \sigma_{-j}}(\sigma_j)}{\sum_{u_j \in \mathcal{U}} B_{\tau, \sigma_{-j}^*}(\sigma_j)^*} \\ &\geq \frac{\mathcal{R}_{min} \sum_{s_i \in S} num_{s_i}(\sigma)}{\mathcal{R}_{max} \sum_{s_i \in S} num_{s_i}(\sigma^*)} \geq \frac{\mathcal{R}_{min}}{\mathcal{R}_{max}} \rho_{user}(\sigma) \end{aligned}$$

Based on Lemma 2, the lower bound of $\rho_{benefit}$ satisfies:

$$\rho_{benefit} \geq \frac{\mathcal{R}_{min}(\lfloor T_{min}/Q_{max} \rfloor)}{\mathcal{R}_{max}(\lfloor T_{max}/Q_{min} \rfloor + 1)}$$

The overall system benefit achieved by DUPA^{3}Game cannot be higher than that achieved by the optimal EDC strategy σ^*. Thus, Theorem 5 holds. □