Joint Communication and Computation Scheduling for MEC-enabled AIGC Services Based on Generative Diffusion Model

Huaizhe Liu, Jiaqi Wu, Xinyi Zhuang, Hongjia Wu, and Lin Gao

***Abstract*—Artificial Intelligence-Generated Content (AIGC) based on Generative Diffusion Model (GDM) has emerged as a promising paradigm of content generation, revolutionizing the creation of diverse contents and driving significant technological advancements. Due to the low latency requirements of AIGC services, mobile edge computing (MEC) has become a crucial enabling technology for these services. In this work, we consider an MEC-enabled GDM-based AIGC network, which consists of multiple GDMs with varying sizes and capabilities deployed on edge computing servers (ES), and multiple mobile users (UEs) with diverse latency and accuracy requirements requesting AIGC services from ES through wireless access points (APs). In such a scenario, we are interested in the joint communication and computation scheduling problem for UEs, which involves selecting the appropriate APs (along with the communication bandwidth allocation) and the appropriate ES (together with the computation resource allocation and model inference optimization) for UEs, considering both the UEs' heterogeneous requirements and the GDMs' heterogeneous capabilities. To address the problem in a practical scenario with decentralized, autonomous, and self-interested UEs, we formulate a non-cooperative game, called the *Joint User Association and Computation Offloading (JUACO)* game, where each UE acts as a game player, selecting the best AP (for communication) as well as the best ES and the best GDM model inference step (for computation), aiming to minimize the inference time while meeting the specified inference accuracy requirement. We prove that the proposed JUACO game is a *potential game*, thus guaranteeing the existence of Nash equilibrium (NE) and the convergence of simple best response-based distributed algorithms to NE. Simulation results demonstrate that the proposed JUACO game approach significantly reduces the inference time and meets the required accuracy compared to traditional methods, validating the effectiveness and practicality of the game-theoretic approach in real-world scenarios.**

***Index Terms*—Artificial Intelligence-Generated Content, Generative Diffusion Model, Mobile Edge Computing**

I. Introduction

A. Background and Motivations

Nowadays, AI-Generated Content (AIGC) based on Generative Diffusion Models (GDMs) has emerged as a promising paradigm of content generation, revolutionizing the creation of diverse contents and driving significant technological advancements. GDMs utilize a stochastic denoising process to transform simple noise distributions into complex data distributions [1]. This process happens gradually through a series of small, reversible *inference steps*. At each step, GDMs predict the noise component that needs to be subtracted to move towards the original data distribution. Due to the scalability with large datasets and complex data distributions, GDMs are well-suited for various tasks across different domains, such as image synthesis and text generation [2].

Based on the above, GDMs generate contents through an iterative process that gradually denoise a sample starting from pure noise. Each inference step in this process involves computing a reverse diffusion process, which is computationally intensive and time-consuming. Several research efforts have been made to mitigate the complexity of GDMs. For example, in [3] and [4], H. Chung *et al.* and R. Rombach *et al.* modified the inference mechanisms to enhance image generation performance while significantly reducing the inference steps. In [5], Xu *et al.* proposed a multi-task multimodal network named Versatile Diffusion (VD) to reduce the computational complexity. However, due to the huge model size and high computational complexity, GDMs are often deployed on remote cloud servers. Therefore, transmitting the generated contents (e.g., image and video) back to users may also consume a significant amount of communication resource, resulting in additional transmission delay.

To this end, *Mobile Edge Computing* (MEC) [6]–[8] has become a crucial enabling technology for these GDM-based AIGC services, attracting widespread attention from many researchers (e.g., [9]–[13]). Existing works in the field of MEC-enabled GDMs mainly focused on the service managing, model caching, and computation offloading, aiming to improve the quality-of-experience (QoE) of users. For instance, Fan *et al.* in [9] proposed a decentralized framework for delivering mobile GDM-based AIGC services to users to strike a balance between the supply of AIGC services and users' demand in the Internet of Vehicles (IoV) scenario. Du *et al.* in [10] proposed an AIGC-as-a-Service (AaaS) scheme in MEC networks to enhance the user QoE in GDM-based of AIGC services. Xu *et al.* in [11] developed a joint GDM caching and inference framework to balance the tradeoff between inference latency, accuracy, and computation resource consumption. Wang *et al.* in [13] proposed a joint optimization framework for user

Authors are with the School of Electronics and Information Engineering and the Guangdong Provincial Key Laboratory of Aerospace Communication and Networking Technology, Harbin Institute of Technology, Shenzhen, China. (*Corresponding Author: Lin Gao*) Email: gaol@hit.edu.cn.

This work is supported in part by National Key Research and Development Program of China (Grant No. 2021YFB2900300), National Natural Science Foundation of China (Grant No. 61972113), Shenzhen Science and Technology Program (Grant No. KQTD20190929172545139, GXWD20231129103946001, and ZDSYS20210623091808025), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2021B1515120008), and Natural Science Foundation of Guangdong Province (Grant No. 2024A1515010178).

ISBN 978-3-903176-65-2 © 2024 IFIP

association, computation offloading, and inference steps adjustment of GDM-based AIGC tasks.

However, most of the aforementioned studies didn't account for the heterogeneity of GDMs (e.g., [9], [10], [12], [13]) or the diverse QoE requirements of users (e.g., [9]–[11]). In practice, different GDMs may have different model sizes and capabilities, result in different levels of service quality. Moreover, many existing studies in this field focused on the sole optimization of computation resource (e.g., [10], [11]) or communication resource (e.g., [12]), without considering the joint optimization of both resources. Finally, many existing studies proposed centralized approaches (e.g., [10], [11], [13]), which may not be suitable for practical scenarios with decentralized, autonomous, and self-interested users. To address these issues, this work considers a general GDM-based AIGC network scenario that includes heterogeneous GDMs and diverse user requirements. Our primary goal is to investigate the joint optimization of communication and computation resources in a decentralized manner.

B. Solution and Contributions

Specifically, we consider a general MEC-enabled mobile AIGC network scenario as shown in Fig.1, which consists of multiple edge computing servers (ES) offering GDM-based AIGC services, wireless access points (APs) providing communication services, and mobile users (UEs) requesting AIGC services from ES via APs. Each ES is deployed with a specific GDM with a particular size and capability. Different GDMs (on different ES) may have different sizes and capabilities, thus can provide AIGC services with different qualities. APs are located at different locations and can provide communication services for UEs in a given coverage area with a limited communication bandwidth. UEs are running different AIGC applications with diverse latency and accuracy requirements, and can request AIGC services from different ES through different APs, considering the GDMs' different capabilities and the APs' different locations.

As shown in Fig.1, UEs 1 and 2 are running the text generation applications (e.g., chatGPT[1]), UEs 3 and 4 are running the picture generation applications (e.g., Midjourney[2]), and UEs 5 and 6 are running the video generation applications (e.g., ImagineArt[3]). ES 1, 2, and 3 are deployed with different GDMs, which have different capabilities in generating different contents. In this example, UE 1 accesses network via AP 1, and requests AIGC service from ES 3. UE 5 accesses network via AP 2, and request AIGC service from ES 1, and so on. This example also illustrates that the quality of generated images increases with the inference steps.

More specifically, when a UE initiates a service request, it needs to choose an AP to access network, and an ES (or GDM) to generate contents. Note that those UEs choosing the same AP or ES will compete with each other for the limited communication bandwidth or computation resource. Moreover,

[1] https://openai.com/chatgpt/
[2] https://www.midjourney.com/
[3] https://www.imagine.art/

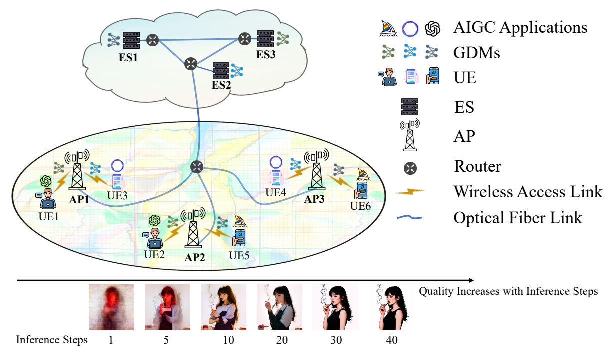


Fig. 1. An example of MEC-enabled GDM-based AIGC Network.

as demonstrated in many existing studies (e.g., [13], [14]), the quality of content generated by GDMs is highly dependent on the number of inference steps. Specifically, a larger number of inference steps results in higher quality but also incurs a higher computational cost, as illustrated in Fig. 1. Therefore, UEs must carefully determine the number of inference steps to balance content quality and computational cost. Based on the above discussion, in this work we are interested in the following problems:

1) *UE-AP Association Problem*: How to associate each UE with the most appropriate AP, considering the different locations of APs as well as the varying communication loads on APs?
2) *Computation Offloading Problem*: How to offload each UE's task to the most appropriate ES, and what is the optimal inference step, considering the differing GDMs on ES and the varying computation burdens on ES?

The problems are particularly challenging in a practical scenario with decentralized, autonomous, and self-interested UEs, and game theory [15]–[17] becomes a powerful tool to analyze and predict the behaviors of UEs. To this end, we formulate a non-cooperative game, called the *Joint User Association and Computation Offloading (JUACO)* game, where each UE acts as a game player, selecting the best AP (for communication) as well as the best ES together with the best inference step (for computation), aiming to minimize the inference time while meeting the specified inference accuracy requirement. We prove that the proposed JUACO game is a *potential game*, thus guaranteeing the existence of Nash equilibrium (NE) and the convergence of simple best response-based distributed algorithms to NE. Finally, by leveraging the finite improvement property of potential games, we develop a distributed algorithm that converges to NE. To sum up, the main contributions of this work are summarized as follows:

- *Novel Model Scenario:* We consider a novel MEC-enabled GDM-based AIGC network scenarios, which consists of heterogeneous GDMs with varying sizes and capabilities and AIGC users with diverse QoE requirements. We study the joint communication and computation scheduling problem for UEs in a practical scenario with decentralized, autonomous, and self-interested UEs.
- *Novel Game-Theoretic Analysis:* We formulate a non-cooperative game, where each UE acts as a game player,

making the best communication and computation decisions, aiming to minimize the inference delay cost while meeting the specified inference accuracy requirement. We prove that the proposed game is a *potential game*, thus guaranteeing the existence of NE. We further propose distributed algorithms that efficiently converges to NE.

- *Performance Evaluations and Insights:* Simulation results demonstrate that the proposed game approach significantly reduces the inference time while meeting the required accuracy, compared to traditional methods. This validates the effectiveness and practicality of the game-theoretic approach in real-world scenarios.

The rest of the paper is organized as follows. In Section II, we present the system model. In Section III, we formulate and analyze the JUACO game. In Section IV, we present simulation results. Finally, we conclude in Section V.

II. System Model

A. Network Model

As shown in Fig.1, we consider an MEC network consisting of a set $\mathcal{M} = \{1, 2, ..., M\}$ of wireless APs, a set $\mathcal{K} = \{1, 2, ..., K\}$ of ES, and a set $\mathcal{N} = \{1, 2, ..., N\}$ of UEs. Each AP is located at a specific location and has a given wireless signal coverage area, providing network access service for UEs within its signal coverage area. Each ES is deployed with a trained GDM and can provide GDM-based AIGC. For notational convenience, we will refer to the GDM deployed on ES k as GDM k, if no confusion is caused. AP and ES are connected with routers and switches via optical fiber links. Each UE is running a specific AIGC application, and can access network via a nearby AP through wireless link to request AIGC service from an ES. Different from cloud computing servers often deployed on the remote Internet (e.g., Google Data Centers), ES are deployed at the network edge that approximates to UEs, such as routers and switches, and thus can provide AIGC services for UEs with low latency.

B. GDM-based AIGC Service Model

We consider a general AIGC scenario, where UEs request different AIGC services, and ES deploy different GDMs. Each GDM is capable of providing multiple AIGC services; however, the quality and cost of these services vary depending on the model's capabilities and characteristics. For example, a GDM specialized in natural language processing might offer superior text generation and translation services at a low cost but may not excel in generating image or video content.

The service quality of GDMs are influenced by factors such as the model size, training data, and computational resources required for operation [2], [13]. For instance, larger models with extensive training on diverse datasets tend to produce higher-quality outputs but also demand more computational power and storage, leading to increased operational costs. Conversely, smaller, more specialized models might offer cost-effective solutions for segmented applications but may lack the versatility and performance of their larger counterparts.

As illustrated in [13], the quality of GDM-based AIGC services is evaluated by the *average error of generated contents* (AEC), which is highly dependent on the number of inference steps, i.e., the reverse diffusion steps used by GDMs for iterative denoising. Let E_{nk} denote the AEC of UE n's content generated by GDM k. Let d_{nk} denote the inference steps used by GDM k for UE n's content. According to [13], the AEC of UE n's content generated by GDM k can be evaluated as follows:

$$E_{nk} = \epsilon_k^{\mathrm{fwd}} \cdot e^{-\gamma_{nk} \cdot d_{nk}}, \forall k \in \mathcal{K}, n \in \mathcal{N}, \tag{1}$$

where $\epsilon_k^{\mathrm{fwd}}$ is a scaling factor evaluating the forward process of GDM k, and γ_{nk} is an attenuation factor characterizing the recovering ability of GDM k for the AIGC task of UE n. Clearly, a larger γ_{nk} leads to a smaller AEC E_{nk}, implying that GDM k is more preferable for the AIGC task of UE n. Moreover, different $\gamma_{nk}, \forall n, k$ imply that different GDMs have different recovering abilities for different AIGC contents.

The computation cost of generating an AIGC content by a GDM is measured by the total number of Floating Point Operations (FLOPS), which is proportional to the number of inference steps used by the GDM. Let ξ_k denote the number of FLOPS used by one inference step of GDM k (i.e., FLOPS per step). Clearly, the value of ξ_k is related to the model size and architecture. For example, a larger model GDM k and/or more complicated architecture often lead to a higher ξ_k. Then, the total number of FLOPS used by GDM k to generate UE n's AIGC content, denoted by F_{nk}, can be expressed as:

$$F_{nk} = \xi_k \cdot d_{nk}, \forall k \in \mathcal{K}, n \in \mathcal{N}. \tag{2}$$

Without loss of generality, we assume that different GDMs will generate AIGC contents with the same data size for the same AIGC application. This is reasonable because many AIGC applications, such as text generation, image creation, or audio synthesis, often have standardized output formats and sizes. For example, an image generation application may always produce images of a specific resolution (e.g., 1024×1024 pixels), regardless of the underlying model used. For convenience, we introduce D_n to denote the data size (in MB) of the contents generated for UE n.

C. Communication Model

As shown in Fig.1, when a UE requests an AIGC service, it needs to select an ES or GDM for content generation and an AP for data transmission. Next, we will discuss the communication process, which mainly consists of the wireless communications between UEs and APs and the wired communications between APs and ES.

1) Communication links between UEs and APs: Let R_{nm} denote the channel capacity of the wireless link between UE n and AP m. According to Shannon-Hartley theorem:

$$R_{nm} = W \cdot \log_2(1 + \frac{\rho_{nm} \cdot h_{nm}}{N_0}), \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{3}$$

where ρ_{nm} and h_{nm} denote the transmission power and the wireless channel gain between UE n and AP m, respectively.

Then, the total transmission time required for transmitting the generated content to UE n from AP m is:

$$T_{nm}^{\mathrm{Acc}} \triangleq \frac{D_n}{R_{nm}}, \forall m \in \mathcal{M}, n \in \mathcal{N}. \tag{4}$$

Note that the transmission time in (4) is derived under the assumption that UE n occupies the whole wireless transmission capacity of AP m. In practice, however, when multiple UEs select the same AP, they need to share the limited communication resource (e.g., bandwidth) with each other. In this work, we adopt a simple resource-sharing mechanism based on proportional fairness (PF), where all UEs connected to the same AP and share the total communication resources in proportion to their transmission demands. In such a scenario, all UEs connected to the same AP will complete the data transmission at approximately the same time. Let $x_{nm} \in \{0,1\}$ to denote the UE-AP association indicator, where $x_{nm} = 1$ indicates that UE n is connected to (associated with) AP m. Then, the actual transmitting completion time for each UE connected to AP m can be derived as follows:

$$L_m(\mathbf{X}) = \sum_{n \in \mathcal{N}} x_{nm} \cdot T_{nm}^{\mathrm{Acc}}, \forall m \in \mathcal{M}, \tag{5}$$

where $\mathbf{X} \triangleq (x_{nm}, \forall n, m)$ denotes the UE-AP association relationships between all APs and all UEs. It is easy to see that the above completion time is equivalent to the total time required for AP m to transmit data sequentially to all connected UEs. In this sense, we will also refer to $L_m(\mathbf{X})$ as the *load* of AP m.

Given the UE-AP association vector $\boldsymbol{x}_n = (x_{n1}, ..., x_{nM})$, the actual transmitting completion time of UE n can be written as:

$$T_n^{\mathrm{Acc}}(\mathbf{X}) = \sum_{m \in M} x_{nm} \cdot L_m(\mathbf{X}), \forall n \in \mathcal{N}. \tag{6}$$

2) Communication Links Between APs and ES: As in many existing literature (e.g., [18], [21]), we assume that the wired optical fiber links between APs and ES have infinite capacity. Thus, we ignore transmission time between APs and ES.

D. Computation Model

Now we discuss the computation process of each GDM on each ES. Let f_k denote the total computation capacity of ES k, measured in FLOPS per second (FLOPS/s). Then, the total computation time required by the GDM on ES k for generating content of UE n can be written as:

$$T_{nk}^{\mathrm{Comp}} = \frac{F_{nk}}{f_k}, \forall k \in \mathcal{K}, n \in \mathcal{N}, \tag{7}$$

where $F_{nk} \triangleq \xi_k \cdot d_{nk}$ denotes the inference steps and d_{nk} denotes the total number of FLOPS, used by GDM k to generate UE n's content.

Similar as in (4), the computation time in (7) is derived under the assumption that UE n occupies the whole computation resource of ES k. In practice, however, when multiple UEs select the same ES, they need to share the limited computation resource (i.e., the total FLOPS per second) with each other. Similarly, we adopt a simple PF-based resource-sharing mechanism, where all UEs offloaded to the same ES share the total computation resources in proportion to their computation demands. In such a scenario, all UEs offloaded to the same ES will complete the content generation at approximately the same time. Let $y_{nk} \in \{0,1\}$ denote the computation offloading indicator, where $y_{nk} = 1$ indicates that UE n offloads its task to ES k. Then, the actual computation completion time for each UE connected to ES k can be derived as follows:

$$I_k(\mathbf{Y}, \mathbf{D}) = \sum_{n \in \mathcal{N}} y_{nk} \cdot T_{nk}^{\mathrm{Comp}}, \forall k \in \mathcal{K}, \tag{8}$$

where $\mathbf{Y} \triangleq (y_{nk}, \forall n, k)$ denotes the offloading matrix consisting of the offloading indicators between all ES and all UEs, and $\mathbf{D} \triangleq (d_{nk}, \forall n, k)$ denotes the inference step matrix consisting of the possible inference steps used by all GDMs for all UEs. It is easy to see that the above completion time is equivalent to the total time required for ES k to generate content sequentially for all offloaded UEs. In this sense, we will also refer to $I_k(\mathbf{Y}, \mathbf{D})$ as the *load* of ES k.

Given the offloading vector $\boldsymbol{y}_n = (y_{n1}, ..., y_{nK})$ and the inference step vector $\boldsymbol{d}_n = (d_{n1}, ..., d_{nK})$, the actual computation completion time of UE n can be written as:

$$T_n^{\mathrm{Comp}}(\mathbf{Y}, \mathbf{D}) = \sum_{k \in \mathcal{K}} y_{nk} \cdot I_k(\mathbf{Y}, \mathbf{D}), \forall n \in \mathcal{N}. \tag{9}$$

E. Problem Formulation

Based on (6) and (9), the total service latency for UE n can be written as:

$$T_n^{\mathrm{Total}}(\mathbf{X}, \mathbf{Y}, \mathbf{D}) = T_n^{\mathrm{Acc}}(\mathbf{X}) + T_n^{\mathrm{Comp}}(\mathbf{Y}, \mathbf{D}), \forall n \in \mathcal{N}. \tag{10}$$

The objective is to minimize the total service time of UEs, while meeting the specified inference accuracy requirements of UEs. Therefore, we can formulate the following joint UE association and computation offloading problem $\mathbf{P0}$.

$$\mathbf{P0} \quad \min_{\{\mathbf{X}, \mathbf{Y}, \mathbf{D}\}} \sum_{n \in \mathcal{N}} T_n^{\mathrm{Total}}(\mathbf{X}, \mathbf{Y}, \mathbf{D}) \tag{11}$$

$$\text{s.t.} \sum_{k=1}^{K} y_{nk} \cdot E_{nk} \leq \mathcal{E}_n^0, \forall n \in \mathcal{N}, \tag{11a}$$

$$\sum_{m=1}^{M} x_{nm} = 1, \forall n \in \mathcal{N}, \tag{11b}$$

$$\sum_{k=1}^{K} y_{nk} = 1, \forall n \in \mathcal{N}, \tag{11c}$$

$$x_{nm} \in \{0,1\}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{11d}$$

$$y_{nk} \in \{0,1\}, \forall k \in \mathcal{K}, n \in \mathcal{N}, \tag{11e}$$

$$d_n \in [d^{\mathrm{flr}}, d^{\mathrm{cel}}], d_n \in \mathbb{Z}^+. \tag{11f}$$

In the above problem, (11a) ensures that the achieved AEC for UE n cannot exceed a threshold $\mathcal{E}_n^0$, which essentially characterizes the QoE requirement of UE n; (11b) and (11c) ensure that each UE can only be associated with one AP and be offloaded to one ES. It is easy to see that $\mathbf{P0}$ is a centralized optimization problem, aiming to optimizing the total objective of all UEs, while meeting the QoE requirements of all UEs. This is often unachievable in a practical scenario with decentralized, autonomous, and self-interested UEs. To address the problem in such a scenario, we will formulate a non-cooperative game, where each UE acts as a game player, aiming to optimize its own objective and meet its own QoE requirement.

III. Game Formulation and Analysis

A. Game Formulation

To analyze the problem in a practical scenario with decentralized, autonomous, and self-interested UEs, in this section we will establish a non-cooperative game, where each UE acts as a game player, determining its own AP association, ES selection, and inference steps, aiming to optimize its own objective and meet its own QoE requirement. We refer to such a game as the Joint UE Association and Computation Offloading (JUACO) game.

Game 1 (Joint UE Association and Computation Offloading Game)**.** *The Joint UE Association and Computation Offloading Game (JUACO) game, denoted by $\mathcal{G}^J \triangleq \{\mathcal{N}, \{\boldsymbol{s}_n\}_{n\in\mathcal{N}}, \{T_n^{\text{Total}}(\boldsymbol{s}_n, \mathbf{S}_{-n})\}_{n\in\mathcal{N}}\}$ is defined as follows:*

- ***Players:*** *All UEs $n \in \mathcal{N}$ in the system;*
- ***Strategies:*** *Each UE n selects its strategy $\boldsymbol{s}_n \triangleq (\boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{d}_n)$, where $\boldsymbol{x}_n$ is the UE-AP association strategy, $\boldsymbol{y}_n$ is the offloading strategy, and $\boldsymbol{d}_n$ is the inference step strategy.*
- ***Payoffs:*** *The payoff for UE n is its service completion time, i.e., $T_n^{\text{Total}}(\boldsymbol{s}_n, \mathbf{S}_{-n})$, which depends not only on its strategy $\boldsymbol{s}_n$ but also the strategies of all the other UEs in the system except n, i.e., $\mathbf{S}_{-n} = (\boldsymbol{s}_1, ..., \boldsymbol{s}_{n-1}, \boldsymbol{s}_{n+1}, ..., \boldsymbol{s}_N)$.*

For a non-cooperative game, Nash equilibrium (NE) represents a state where each player's strategy is optimal given the strategies chosen by the other players. Namely, in an NE, no player can improve their own payoff by changing their strategy alone, assuming the others keep theirs unchanged. We give the definition of NE, as follows:

Definition 1 (Nash Equilibrium)**.** *A strategy profile defines an **NE** of $\mathcal{G}^J$, if for every player $n \in \mathcal{N}$, we have*

$$T_n^{\text{Total}}(\boldsymbol{s}_n^*, \mathbf{S}_{-n}^*) \leq T_n^{\text{Total}}(\boldsymbol{s}_n, \mathbf{S}_{-n}^*), \tag{12}$$

where $\mathbf{S}_{-n}^*$ denotes the strategy of all the other players except player n, i.e., $\mathbf{S}^* = (\boldsymbol{s}_1^*, \ldots, \boldsymbol{s}_n^*, \ldots, \boldsymbol{s}_N^*)$.

We next show that the JUACO game can be decoupled into two independent game, namely, the UE Association (UA) game and the Computation Offloading (CO) game. Therefore, the NE of the JUACO game can be derived by finding the NE of the UA game and the CO game.

Game 2 (UE Association Game)**.** *The UE Association Game (UA) game, denoted by $\mathcal{G}^A \triangleq \{\mathcal{N}, \{\boldsymbol{x}_n\}_{n\in\mathcal{N}}, \{T_n^{\text{Acc}}(\boldsymbol{x}_n, \mathbf{X}_{-n})\}_{n\in\mathcal{N}}\}$, is defined as follows:*

- ***Players:*** *All UEs $n \in \mathcal{N}$ in the system;*
- ***Strategies:*** *Each UE n selects its UE-AP association strategy $\boldsymbol{x}_n$.*
- ***Payoffs:*** *The payoff for UE n is its transmitting completion time, i.e., $T_n^{\text{Acc}}(\boldsymbol{x}_n, \mathbf{X}_{-n})$, which depends not only on its UE-AP association strategy $\boldsymbol{x}_n$ but also the strategies of all the other UEs in the system except UE n, i.e., $\mathbf{X}_{-n} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{n-1}, \boldsymbol{x}_{n+1}, ..., \boldsymbol{x}_N)$.*

Game 3 (Computation Offloading Game)**.** *The Computation Offloading Game (CO) game, denoted by $\mathcal{G}^C \triangleq \{\mathcal{N}, \{\boldsymbol{z}_n\}_{n\in\mathcal{N}}, \{T_n^{\text{Comp}}(\boldsymbol{z}_n, \mathbf{Z}_{-n})\}_{n\in\mathcal{N}}\}$ is defined as follows:*

- ***Players:*** *All UEs $n \in \mathcal{N}$ in the system;*
- ***Strategies:*** *Each UE n selects its strategy $\boldsymbol{z}_n \triangleq (\boldsymbol{y}_n, \boldsymbol{d}_n)$, where $\boldsymbol{y}_n$ is the computation offloading strategy, $\boldsymbol{d}_n$ is the inference step strategy.*
- ***Payoffs:*** *The payoff for UE n is its computation completion time, i.e., $T_n^{\text{Comp}}(\boldsymbol{z}_n, \mathbf{Z}_{-n})$, which depends not only on its strategy $\boldsymbol{z}_n$, but also the strategies of all the other UEs in the system except UE n, i.e., $\mathbf{Z}_{-n} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_{n-1}, \boldsymbol{z}_{n+1}, ..., \boldsymbol{z}_N)$.*

In the next subsections, we will show that both the UA game and the CO game are potential games.

B. Analysis of UE Association Game

In this subsection, we analyze NE of the UA game. We will prove that it is a potential game, which always possess NE. To begin, we first provide the definition of potential game.

Definition 2 (Potential Game)**.** *A game is a potential game if and only if there exists a potential function $\Phi(\mathbf{X})$ such that:*

$$T_n^{\text{Acc}}(\boldsymbol{x}_n', \mathbf{X}_{-n}) < T_n^{\text{Acc}}(\boldsymbol{x}_n, \mathbf{X}_{-n}) \Leftrightarrow \Phi(\boldsymbol{x}_n', \mathbf{X}_{-n}) < \Phi(\boldsymbol{x}_n, \mathbf{X}_{-n}) \quad \forall n \in \mathcal{N}, \boldsymbol{x}_n, \boldsymbol{x}_n' \in \mathbf{X}. \tag{13}$$

Eq.(13) implies that, in a potential game, the change in the potential function Φ, due to a unilateral strategy deviation, must share the same sign as the change in the payoff function.

Next, we prove that the UA game is a potential game.[4]

Theorem 1. *The UA game is a potential game with the following potential function:*

$$\Phi(\boldsymbol{x}_n, \mathbf{X}_{-n}) = \sum_{m=1}^{M} \varphi^{L_m(\mathbf{X})}, \tag{14}$$

where $\varphi \geq \varphi_0$ is any value larger than a threshold φ_0, and the threshold φ_0 is a positive constant related to the minimum time scale of communication systems.

Lemma 1. *The UA game has at least one Nash equilibrium.*

Intuitively, at the optimal solution $(\boldsymbol{x}_n^*, \mathbf{X}_{-n}^*)$, the potential function cannot be further minimized. According to (13), the payoff of each game player can neither be further improved, which implies that the optimal strategy is a Nash equilibrium.

Lemma 2. *The best response iteration algorithm will always converge to a Nash equilibrium of the UA game.*

[4]Due to space limit, we put detailed proof in online technical report [22].

The main idea is that the best response (BR) of any player will improve its payoff (i.e., minimize T_n^{Acc}), thereby accordingly decreasing the potential function. Since the potential function is bounded, the aforementioned process converges within a finite number of steps. By leveraging the finite improvement property (FIP) of potential games, we can further demonstrate the convergence of classic distributed algorithms, such as BR iterations, toward a Nash equilibrium.

C. Analysis of Computation Offloading Game

In this subsection, we analyze the NE of the CO game. To proceed, we first give the potential function for the CO game and hence prove that the CO game is a potential game.

Theorem 2. *The CO game is a potential game with the following potential function $\Psi(\boldsymbol{z}_n, \mathbf{Z}_{-n})$:*

$$\Psi(\boldsymbol{z}_n, \mathbf{Z}_{-n}) = \sum_{k=1}^{K} \phi^{I_k(\mathbf{Y}, \mathbf{D})}, \tag{15}$$

where $\phi \geq \phi_0$ is any value larger than a threshold ϕ_0, and the threshold ϕ_0 is a positive constant related to the minimum time scale of computation systems.

Similar to the analysis for the UA game in the previous subsection, we can demonstrate that the CO game possesses at least one NE. Furthermore, it can also be confirmed that the BR iterative algorithm consistently converges to the NE of the CO game.

So far, we have demonstrated that both the UA game and the CO game are potential games that always possess NE. Therefore, achieving the NE of both the UA game and the CO game implies that the NE of the JUACO game can also be achieved. In the following subsection, we will design two distributed algorithms based on the best response iteration for the UA game and the CO game, aiming to achieve the NE.

D. Algorithm Design

The detailed algorithm design for the UA game and the CO game are given in Algorithm 1 and Algorithm 2, respectively. To accelerate the convergence, we allow multiple players (UEs) to update strategies simultaneously in each round of iteration (see line 4 of Algorithm 1 and Algorithm 2), instead of choosing only one player as in the traditional best response iteration algorithm.

IV. Numerical Results

In this section, we perform simulations to evaluate the performance of the distributed JUACO approach. In the simulations, we consider a 1km$\times$1km square area, where 5 APs are distributed evenly and UEs are randomly distributed. The other simulation parameters are presented in Table I. Similar experimental configurations are adopted in [11], [13].

A. Convergence

We first illustrate the individual service delay dynamic of each UE with the iteration rounds when the UE number is 40. From Fig.2(a) and Fig.2(b), we can observe that both the transmition completion time and the computation completion time of UE gradually become stable. This implies that the algorithm converges to a stable point, indicating that the NE of both the UA game and the CO game are acheived. Consequently, the NE of the JUACO game is hence reached.

Algorithm 1: BR Based Distributed Algorithm for $\mathcal{G}^A$

Input: Network parameters in Table I
Output: UE-AP strategy of each UE, $\boldsymbol{x}_n, n \in \mathcal{N}$

1: **Initialize** $\boldsymbol{x}_n, \forall n \in \mathcal{N}$
2: **while** not converged **do**
3: $\quad t = t + 1$
4: $\quad$ Randomly select a subset of $\mathcal{N}_t \in \mathcal{N}$
5: $\quad$ **for** each $n \in \mathcal{N}_t$ in parallel **do**
6: $\quad\quad \boldsymbol{x}_n^* = \arg\min_{\boldsymbol{x}_n} T_n^{\mathrm{Acc}}(\boldsymbol{x}, \mathbf{X}_{-n}^{t-1})$
7: $\quad\quad \boldsymbol{x}_n^t \longleftarrow \boldsymbol{x}_n^*$
8: $\quad$ **end for**
9: $\quad$ **if** $\boldsymbol{x}_n^t = \boldsymbol{x}_n^{t-1}$ **then**
10: $\quad\quad$ **Break**
11: $\quad$ **end if**
12: **end while**

Algorithm 2: BR Based Distributed Algorithm for $\mathcal{G}^C$

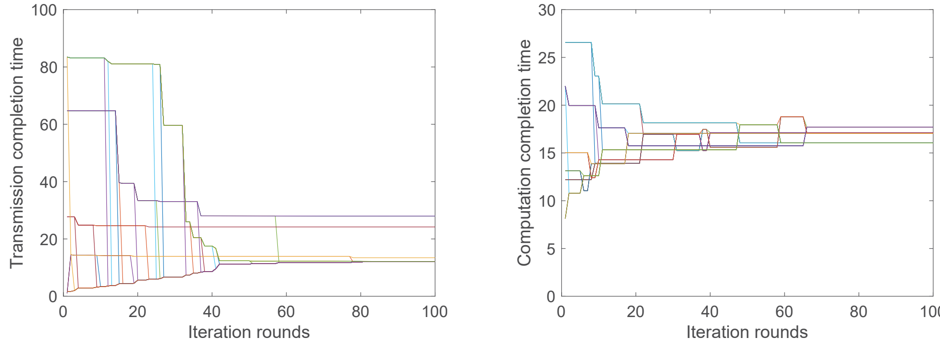
Input: Network parameters in Table I
Output: UE-AP strategy of each UE, $\boldsymbol{z}_n, n \in \mathcal{N}$

1: **Initialize** $\boldsymbol{z}_n, \forall n \in \mathcal{N}$
2: **while** not converged **do**
3: $\quad t = t + 1$
4: $\quad$ Randomly select a subset of $\mathcal{N}_t \in \mathcal{N}$
5: $\quad$ **for** each $n \in \mathcal{N}_t$ in parallel **do**
6: $\quad\quad \boldsymbol{z}_n^* = \arg\min_{\boldsymbol{z}_n} T_n^{\mathrm{Comp}}(\boldsymbol{z}, \mathbf{Z}_{-n}^{t-1})$
7: $\quad\quad \boldsymbol{z}_n^t \longleftarrow \boldsymbol{z}_n^*$
8: $\quad$ **end for**
9: $\quad$ **if** $\boldsymbol{z}_n^t = \boldsymbol{z}_n^{t-1}$ **then**
10: $\quad\quad$ **Break**
11: $\quad$ **end if**
12: **end while**

B. Performance Comparisons with Baseline Solutions

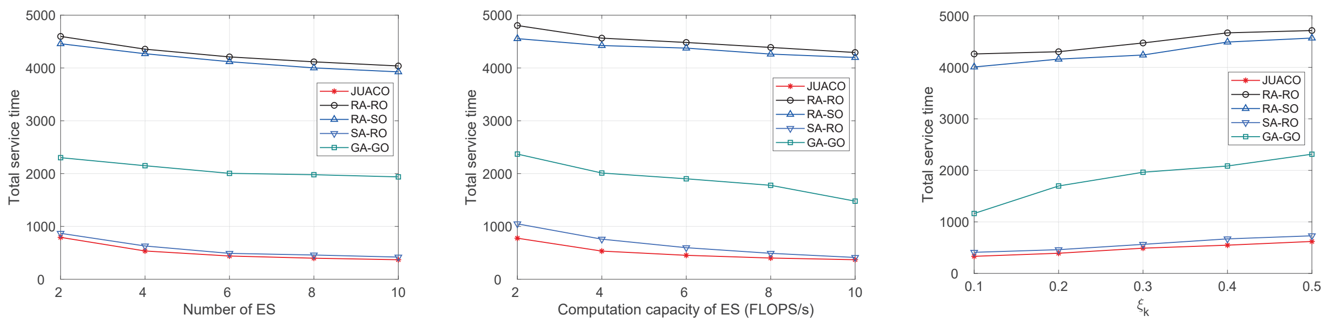
We compare our proposed game-based JUACO solution with the following baseline solutions, that is:

1) *Random UE Association and Random computation Offloading (RA-RO Solution)*: UEs associate with APs and select ES randomly.
2) *Random UE Association and Strategic Offloading (RA-SO Solution)*: UEs randomly associate with APs but adopt the BR based algorithm Alg.(2) to select ES for task offloading strategically.
3) *Strategic UE Association and Random Offloading (SA-RO Solution)*: UEs adopt the BR based algorithm Alg.(1) to select AP strategically for data transmission while randomly select ES for task offloading.
4) *Greedy Association and Greedy Offloading (GA-GO Solution)* [21]: UEs select the nearest AP for data transmission and choose the ES equipped with the largest



(a) (b)

Fig. 2. Dynamics of UEs' service time (a) Transmission completion time, and (b) Computation completion time



(a) (b) (c)

Fig. 3. Total service time vs (a) ES Number, (b) Computation capacity of ES, and (c) The number of FLOPS used by one inference step

TABLE I
MAIN SIMULATION PARAMETERS

Parameter	Value
Number of APs M	5
Number of ES K	[2,10]
Number of UEs N	[5,30]
FLOPS used by one inference step ξ_k	[0.1,0.5] TeraFLOPS/step
Computation capacity of ES f_k	[2,10] TeraFLOPS
UE transmission data volume D_n	10 MB
Wireless bandwidth of AP W	[2,10] MHz
Transmission power of each UE ρ_{nm}	0.2 w
Path loss factor θ	4
The back-ground Noise Power N_0	-174 dBm/Hz

computation capacity for task processing, regardless the others strategies.

The Impact of Strategies. From Fig.3-5, we can observe that the RA-RO solution always attains the highest total service time, compared to the other four solutions. This is due to the ignoring of both the UE association strategy and the computation offloading strategy of UE. The RA-SO solution considers only the computation task offloading strategy, while the SA-RO solution considers only the UE association strategy. Consequently, both the RA-SO and SA-RO solutions outperform the RA-RO solution. Besides, the performance of the SA-RO solution outperforms the RA-SO solution because the proportion of transmission completion time to the total service time is higher than that of computation completion time to the total service time.

In both the GA-GO solution and our proposed JUACO solution, UEs employ rational strategies. In the GA-GO scheme, UEs select the AP nearest to them and the ES with the best computation resources, which gradually leads to the degradation performance of these APs and ES, thereby increasing latency for each UE. In contrast, in our proposed JUACO solution, UEs consider the decisions and utilities of each other when choosing APs and ES, potentially leading to the best response solution at the NE point.

More importantly, it is shown in Fig.3-5 that our proposed JUACO solution alway achieves the best performance in minimizing the total service time.

The Impact of ES. Fig.3 illustrates the impact of ES parameters. From Fig.3(a), we can see that the total service time decreases with the ES number. This is because as the number of ES increases, the number of UEs serviced by each ES decreases, leading to reduced computation completion time for UEs. Fig.3(b) shows that the total service time decreases with the computation capacity of ES, since a larger computation capacity leads to a lower completion time. Fig.3(c) shows that the total service time increases with the number of FLOPS consumed by one inference step. This is because the increase in ξ_k brings larger computation burden.

The Impact of AP. From Fig.4 we can see that the total service time decrease with the wireless bandwith of AP for all five solutions. This is because a larger bandwith can shorten the UE data transmission completion time and hence lead to a smaller total service time.

The Impact of UE. Fig.5 shows that the total service time

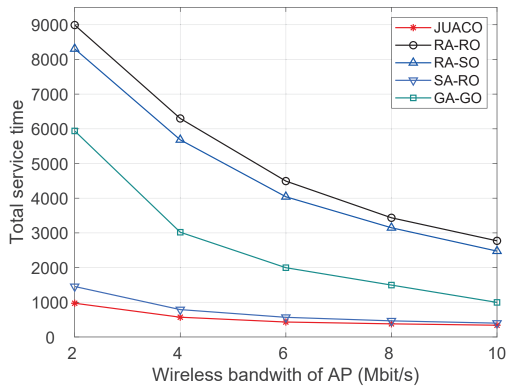


Fig. 4. Total service time vs wireless bandwith of AP.

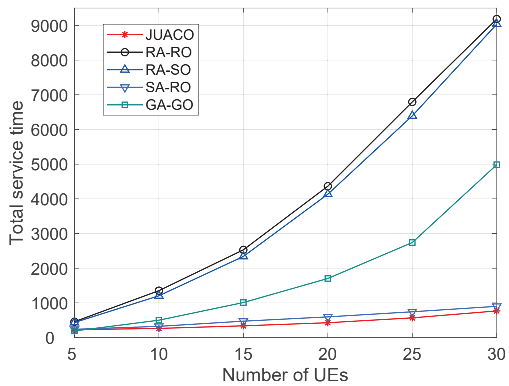


Fig. 5. Total service time vs the number of UEs.

increases with the number of UEs for all five solutions. This is because more UEs implies a heavier transmission burden on each AP and a heavier computation burden on each ES and hence leads to a higher transmission completion time and a higher computation completion time. Furthermore, we can see that as the number of UEs increases, the performance acheived by the RA-RO, RA-SO, SA-RO, and GA-GO solutions deteriorates more rapidly compared to out proposed JUACO solution. This implies that our proposed JUACO solution exhibits more stable performance when faced with a surge in UE numbers.

V. Conclusion

In this work, we study the joint communication and computation scheduling problem in a general MEC-enabled GDM-based AIGC network. We focus on the joint optimization of UE-AP association and computation task offloading problem, which involves selecting the appropriate APs and ES, as well as optimizing model inference steps for UEs, under the consideration of UEs' heterogeneous requirements and the heterogeneity of GDMs' capacities. Considering the UEs' self-interest nature, we further formulate the problem as a non-cooperative game. We further prove that this game is a potential game, wich ensures the existence of Nash equilibrium. We develop a distributed algorithm based on the best response strategy that iteratively achieves a Nash equilibrium. Numerical simulations demonstrate that our proposed solution not only significantly reduces inference time but also achieves the required accuracy, surpassing traditional methods. This outcome validates the effectiveness and practicality of the game-theoretic approach in real-world scenarios.

References

[1] M. Xu, H. Du, D. Niyato, J. Kang, and J. Xiong, et al., "Unleashing the Power of Edge-Cloud Generative AI in Mobile Networks: A Survey of AIGC Services," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1127-1170, Secondquarter 2024.

[2] F. -A. Croitoru, V. Hondru, R. T. Ionescu and M. Shah, "Diffusion Models in Vision: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 10850-10869, Sept. 2023.

[3] H. Chung, B. Sim, and J. C. Ye, "Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems Through Stochastic Contraction," *Proc. IEEE CVPR*, 2022, pp. 12 413–12 422.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," *Proc. IEEE CVPR*, 2022, pp. 10 684–10 695.

[5] X. Xu, Z. Wang, E. Zhang, K. Wang and H. Shi, "Versatile Diffusion: Text, Images and Variations All in One Diffusion Model," *Proc. IEEE ICCV*, 2023, pp. 7720-7731.

[6] Z. Wang, L. Gao, T. Wang, and J. Luo, "Monetizing Edge Service in Mobile Internet Ecosystem," *IEEE Trans. Mob. Comput.*, vol.21, no.5, pp.1751-1765, 2022.

[7] C. Jiang, Z. Luo, L. Gao, and J. Li, "A Truthful Incentive Mechanism for Movement-Aware Task Offloading in Crowdsourced Mobile Edge Computing Systems," *IEEE Int. Things J.*, vol.11, no.10, 2024.

[8] Z. Chen, Q. Ma, L. Gao, and X. Chen, "Price Competition in Multi-Server Edge Computing Networks under SAA and SIQ Models," *IEEE Trans. Mob. Comput.*, vol.23, no.1, pp.754-768, 2024.

[9] J. Fan, M. Xu, Z. Liu, H. Ye, C. Gu, D. Niyato, and K. Lam, "A Learning-based Incentive Mechanism for Mobile AIGC Service in Decentralized Internet of Vehicles," *Proc. IEEE VTC*, pp. 1-5, 2023.

[10] H. Du, Z. Li, N. Dusit, *et al.*,"Diffusion-based Reinforcement Learning for Edge-enabled AI-Generated Content Services," *IEEE Trans. Mob. Comput.*, pp. 1-16, Jan. 2024.

[11] M. Xu, D. Niyato, H. Zhang, J. Kang, Z. Xiong, et al., "Joint Foundation Model Caching and Inference of Generative AI Services for Edge Intelligence," *Proc. IEEE GLOBECOM*, pp. 1-6, 2023.

[12] Y. Lin, Z. Gao, H. Du, D. Niyato, J. Kang, et al., "Blockchain-based Efficient and Trustworthy AIGC Services in Metaverse," *IEEE Trans. Serv. Comput.*, pp. 1-13, Jan. 2024.

[13] Y. Wang, C. Liu, and J. Zhao, "Offloading and Quality Control for AI Generated Content Services in Edge Computing Networks," *arXiv preprint*, arXiv:2312.06203.

[14] J. Wang, H. Du, D. Niyato, et al., "A unified framework for guiding generative AI with wireless perception in resource constrained mobile edge networks," *IEEE Trans. Mob. Comput.*, Early Access, 2024.

[15] N. Ding, L. Gao, and J. Huang, "Optimal Pricing Design for Coordinated and Uncoordinated IoT Networks," *IEEE Trans. Mob. Comput.*, vol.22, no.12, pp.7121-7137, 2023.

[16] C, Jiang, L. Gao, J. Luo, P. Zhou, and J. Li, "A Game-Theoretic Analysis of Joint Mobile Edge Caching and Peer Content Sharing," *IEEE Trans. Net. Sci. Eng.*, vol.10, no.3, pp.1445-1461, 2023.

[17] C. Jiang, L. Gao, F. Hou, and J. Li, "Economic Analysis of Edge Caching Enabled Mobile Internet Ecosystem," *IEEE Trans. Mob. Comput.*, Early Access, 2024.

[18] M. Tang, and V. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 1985-1997, Jun. 2022.

[19] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

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

[21] X. Wu, R. Srikant and J. R. Perkins, "Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks," in *IEEE Trans. Mob. Comput.*, vol. 6, no. 6, pp. 595-605, June 2007.

[22] Online Technical Report on OneDrive, available at: https://1drv.ms/b/s!Ast2m14ugp6smc9R2T7INy8CEXEBzw?e=XhbIz8