Joint Mobile Edge Caching and Pricing: A Mean-Field Game Approach

Yin Xu*†, Xichong Zhang*†, Mingjun Xiao*†, Jie Wu‡, An Liu§, and Sheng Zhang¶

*School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

†Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, China

‡Department of Computer and Information Sciences, Temple University, Philadelphia, USA

§School of Computer Science and Technology, Soochow University, Suzhou, China

¶State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Email: {xuyin218@mail.,yyu18@mail.,xiaomj@}ustc.edu.cn, jiewu@temple.edu, anliu@suda.edu.cn, sheng@nju.edu.cn

***Abstract*—In this paper, we investigate the competitive content placement problem in Mobile Edge Caching (MEC) systems, where Edge Data Providers (EDPs) cache appropriate contents and trade them with requesters at a suitable price. Most of the existing works ignore the complicated strategic and economic interplay between content caching, pricing, and content sharing. Therefore, we propose a joint Mean-Field Game framework for mobile edge Caching and Pricing (MFG-CP) in large-scale dynamic MEC systems, which can facilitate distributed optimal decision-making based on the mean-field game theory. Specifically, we first formulate the competitive content placement issue among EDPs as a non-cooperative stochastic differential game. To significantly reduce the communication and computation complexity, we further devise a mean-field model to approximate the collective impact of all EDPs on caching, trading, and sharing, by which each EDP can quickly estimate some unknown information without considerable interactions. Then, we develop a distributed best response scheme based on iterative learning, enabling each EDP to solely customize its optimal caching strategy and pricing policy. Besides, we theoretically prove the existence of a unique MFG equilibrium. Finally, trace-driven simulations demonstrate the effectiveness of MFG-CP compared with some baselines.**

***Index Terms*—Mobile edge caching, dynamic pricing, mean-field game, network economics.**

I. Introduction

A. Background and Motivation

With the proliferation of intelligent devices and services, Mobile Edge Caching (MEC) has become an attractive paradigm for handling the explosive growth of mobile data traffic [1]–[8]. A typical MEC system consists of many Edge Data Providers (EDPs, e.g., small-cell/femtocell base stations and smartphones) and a group of content requesters, through which EDPs can cache appropriate contents (e.g., articles, videos, and music) and then trade them with requesters at a suitable price [9], [10]. Due to its ability to effectively enhance Quality of Experience (QoE) for requesters and reduce bandwidth consumption, MEC has garnered attention from various commercial platforms, e.g., Google Edge Network [11], Fastly [12], CacheMire [13], and Vbrick Edge Caching [14].

In this paper, we focus on the competitive content placement issue in MEC systems, where each EDP not only needs to place proper contents but also needs a pricing policy for content trading. Although much effort has been devoted to content placement [15]–[18], most of them do not discuss the economic competition in caching optimization. Actually, there exists a complex game among EDPs. For example, in the same geographical area, there are two edge video providers (Alice and Bob) offering video services $\{v_1, v_2\}$ to requesters. Here, the popularity of v_1 is higher than v_2, and Alice (or Bob) is capable of only caching one video due to the limited storage resources. Considering that high-popularity videos can generate more trading incomes, Alice and Bob generally tend to cache v_1 to improve their utilities (i.e., net profits). However, when both Alice and Bob cache v_1, there arises a competition that impels them to attract more requesters by sequentially lowering their trading prices. Consequently, both Alice and Bob will earn low utilities or even negative utilities. Then, one of them might prefer to select the other video v_2 to enhance its own utility. In large-scale MEC systems, such the competitive content trading among multiple EDPs will make the content placement issue more challenging. Especially, when EDPs do not cache the content required by the corresponding requester, peer content sharing is conducive to improving the caching efficiency [19]. Hence, it is highly significant to explore the economic effect of dynamic pricing for content placement.

B. Challenges

There are several challenges that need to be overcome in addressing the competitive content placement issue, i.e.,

- *Complicated interplay between caching, pricing, and sharing:* the challenge arises from the fact that caching strategies can significantly impact the trading price under the supply-demand principle, and the rewards of content trading conversely affect the decisions of content replacement. Such the intricate mutual influence is critical since redundant content caching may result in market saturation and decrease the profits of EDPs, while insufficient caching will cause a low QoE for requesters and hurt future trading. Meanwhile, the content sharing among EDPs also involves an economic transaction issue, making the interplay more complex.
- *Real-time decision-making with system dynamics:* the network channel conditions may be unstable due to the random mobility of requesters and some unknown interference. On the other hand, the time-varying content service requests in the dynamic trading market reflect the unpredictable content popularity and timeliness demands. It is quite challenging

to incorporate these spatio-temporal dynamics to adaptively make the optimal caching decision for each EDP.

- *Large-scale distributed best response with incomplete information:* EDPs are coupled together since they coexist in a common trading market. That is, there exists a complicated non-cooperative game among EDPs to make distributed caching decisions. Importantly, it is not trivial for an EDP to estimate the detailed information of other EDPs in the dynamic system, as heavy communication and computation overhead will be consumed, especially for large-scale scenarios. Thus, how to efficiently make the best responses for large-scale EDPs in a distributed way should be solved.

Although a few studies have delved into caching optimization with pricing [20]–[25], the price of all EDPs for the same content determined by these works is homogeneous, ignoring the strategic behaviors of EDPs. Several works have studied the strategy determination in dynamic scenarios [26]–[28]. Nevertheless, they focus on optimizing decision processes and do not consider the complex technical and economic game among multiple EDPs. In short, none of the previous research has comprehensively addressed the three challenges.

C. Solution and Contribution

To circumvent the above challenges, we introduce the Mean-Field Game (MFG) theory [29] to reduce a one-to-many game to a one-to-one game and further propose a joint MFG framework for mobile edge Caching and Pricing, namely MFG-CP, where each EDP can optimize its own utility in a distributed manner. Specifically, we first formulate the system dynamics and design the utility function for each EDP, which takes content pricing, content sharing, content placement, and request service delay into consideration simultaneously. Then, we model the competitive content placement problem among multiple EDPs as a non-cooperative stochastic differential game with incomplete information. To significantly mitigate the communication and computation complexity, we design a joint game framework for MEC systems by harnessing the MFG methodology. In this framework, the collective impact of all EDPs can be approximated via the mean-field distribution, by which each EDP can quickly estimate the unknown information related to its utility without considerable interactions. Accordingly, we present an iterative best response learning scheme to determine the optimal caching strategy and the pricing policy distributedly. In a nutshell, our major contributions are summarized as follows:

- We propose an MFG-CP framework for joint edge caching and pricing, which can facilitate optimal decision-making in a decentralized manner based on the MFG methodology. To the best of our knowledge, this is the first endeavor to explore the intricate interplay between strategic caching and monetary trading in large-scale dynamic MEC systems.
- We formulate the caching optimization problem as a non-cooperative stochastic differential game among multiple EDPs. To achieve low complexity, we devise a mean-field estimator to approximate the impact of caching, trading, and sharing without requiring massive information exchange.

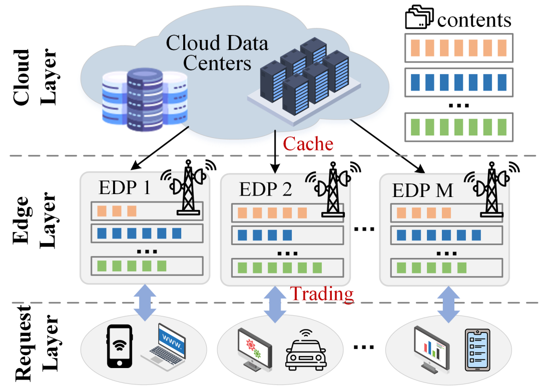


Fig. 1. System overview.

- We develop an iterative best response learning scheme to solve the coupled differential equations. Consequently, each EDP can solely customize the optimal caching strategy and pricing policy using its local information. Additionally, we prove the existence of a unique MFG equilibrium.
- We conduct extensive simulations on the real-world trace to corroborate the significant performance of MFG-CP.

II. System Model

We consider a large-scale MEC system, which consists of a collection of cloud centers, a set $\mathcal{M} = \{1, \cdots, i, \cdots, M\}$ of Edge Data Providers (EDPs), and a group $\mathcal{J} = \{1, \cdots, j, \cdots, J\}$ of content requesters, as illustrated in Fig. 1. Each EDP is endowed with caching functionality to serve requesters who desire some contents. To promptly respond to requesters' demands via wireless links, EDPs need to cache certain contents in advance and are allowed to share the cached contents with each other in a decentralized edge-edge manner. Upon receiving requests, EDPs can offer their cached contents for sale at a suitable trading price. Now, we first present the dynamic network model and the dynamic edge caching model as game foundations and summarize our problem.

A. Network Model

Owing to the randomness and uncertainty of requesters' mobility in the MEC system, there exists some unpredictable interference among requesters, leading to unstable network channel conditions. Consequently, we consider a non-stationary time-varying channel model to describe such a stochastic process. Following the commonly-accepted definition in [27], [30], we let $h_{i,j}(t)$ represent the channel fading coefficient between EDP i and requester j, and the evolution of $h_{i,j}(t)$ can be characterized by a mean-reverting Ornstein-Uhlenbeck process with a dynamic differential equation, i.e.,

$$\mathbf{d}h_{i,j}(t) = \frac{1}{2}\varsigma_h(\upsilon_h - h_{i,j}(t))\mathbf{d}t + \varrho_h \mathbf{d}W_{i,j}(t), \tag{1}$$

where ς_h, υ_h, and ϱ_h are positive constants. ς_h denotes a changing rate, while υ_h and ϱ_h represent the long-term mean and standard deviation of the process, respectively. The random diffusion term $W_{i,j}(t)$ characterizes a standard Brownian motion, capturing the randomness of the system (e.g., the mobility of requesters and channel fluctuation). Eq. (1) depicts

TABLE I
DESCRIPTION OF MAJOR NOTATIONS

Variable	Description
$\mathcal{M}$, i	the set of EDPs and the index of an EDP.
$\mathcal{J}$, j	the set of requesters and the index of a requester.
$\mathcal{K}$, k	the set of contents and the index of a content.
$x_{i,k}(t)$	the caching strategy of EDP i for content k at time t.
$q_{i,k}(t)$	the remaining space of EDP i for content k at time t.
$\Pi_{i,k}$	the content popularity of EDP i for content k.
$I_{i,k}(t)$	the set of requesters who ask for content k at time t.
$L_{i,k}$	the content timeliness of EDP i for content k.
$h_{i,j}(t)$	the channel fading between EDP i and requester j.
$H_{i,j}(t)$	the wireless transmission rate from EDP i to requester j.
$p_{i,k}(t)$	the unit price customized by EDP i for selling content k.
$S_{i,k}(t)$	the state of EDP i with respect to a given content k.
$Q_k, \bar{p}_k$	the data size of content k and the unit sharing price.

the random phenomenon of probability fluctuations around the long-term mean υ_h, i.e., the Ornstein-Uhlenbeck process gravitates towards υ_h at a rate ς_h, while simultaneously experiencing random fluctuations due to the Brownian motion.

In general, each requester is associated with a default serving EDP that is nearest geographically. Here, let $\mathcal{J}_i(t)$ be the set of requesters who are served by EDP i at time t. Since the signals received by requesters from the corresponding EDP may undergo the interference from other links, the achievable wireless transmission rate from EDP i to requester j is:

$$H_{i,j}(t) = B\log_2\big(1+\frac{|g_{i,j}(t)|^2G_i}{\varrho^2+\sum_{i'\neq i}^{M}|g_{i',j}(t)|^2G_{i'}}\big). \quad (2)$$

Here, B is the transmission bandwidth, G_i is the transmission power, and ϱ^2 is the noise power. Generally, the channel gain can be defined by $|g_{i,j}(t)|^2 = |h_{i,j}(t)|^2 d_{i,j}^{-\tau}$, where $d_{i,j}$ means the Euclidean distance between EDP i and requester j, and τ denotes the path loss exponent.

B. Edge Caching Model

The integrated cloud center stores various types of contents, denoted by $\mathcal{K} = \{1,\cdots,k,\cdots,K\}$, each of which will be updated at different frequencies. For example, a content contains the traffic flow data of several important roads (or the financial news of some countries), and then the center may update it every hour (or every day). The data size of content k is denoted by Q_k. After downloading contents, each EDP will conduct the content placement and content trading.

During the content placement phase, each EDP needs a caching strategy to optimize its own utility, as outlined in Section III. Let $\boldsymbol{x_i}(t) = [x_{i,1}(t),\cdots,x_{i,K}(t)]$ represent the caching strategy of EDP i, where $x_{i,k}(t) \in [0,1]$ indicates the instantaneous caching rate of content k, representing the caching proportion when EDP i downloads content k from the data center in time slot t. Correspondingly, the time-varying remaining storage capacity of EDP i for all contents is denoted by $\boldsymbol{q_i}(t) = [q_{i,1}(t),\cdots,q_{i,K}(t)]$, where $q_{i,k}(t)$ signifies the remaining space of EDP i for content k. This indicates that each EDP may cache a portion of content k while reserving some space in the storage. To reasonably describe the dynamics of $\boldsymbol{q_i}(t)$, we introduce several influential factors.

Definition 1 (***Content Popularity***)**.** The content popularity of EDP i refers to the frequency at which the content k is requested, denoted by $\Pi_{i,k}$. Initially, the content popularity typically conforms to the Zipf's distribution [31], expressed as $\Pi_{i,k}(t_0) = \frac{1/k^{\iota}}{\sum_{k=1}^{K}1/k^{\iota}}$. Here, $\iota > 0$ signifies the steepness of the distribution. Let $I_{i,k}(t) \in \mathcal{J}_i(t)$ be the set of requesters who ask for content k at time t. Based on the number of requests changing over time, the content popularity can be updated as:

$$\Pi_{i,k}(t) = \frac{K\cdot\Pi_{i,k}(t_0)+|I_{i,k}(t)|}{K+\sum_{k=1}^{K}|I_{i,k}(t)|}. \quad (3)$$

Definition 2 (***Content Timeliness***)**.** To capture the urgency of requesters' demands, we define the content timeliness of EDP i as the level of urgency with which requesters acquire content k, denoted by $L_{i,k} \in [0, L_{max}]$. A larger $L_{i,k}$ means that requesters desire to receive content k with less time delay, e.g., most drivers hope to obtain traffic data as soon as possible for route planning. Suppose each requester $j \in I_{i,k}(t)$ can specify its data timeliness requirement $L_{i,k,j}$ when making a request. At time t, $L_{i,k}(t)$ can be approximated by the average value, i.e., $L_{i,k}(t) = \sum_{j\in I_{i,k}(t)} L_{i,k,j}/|I_{i,k}(t)|$.

Evidently, the remaining space $q_{i,k}(t)$ fluctuates with the caching strategy. Simultaneously, when taking the content popularity into account, EDP i may opt to discard content k if it is rarely requested. On the other hand, from the perspective of response delay, EDP i is inclined to allocate more storage space to content k when it is urgently requested, thereby mitigating the need for additional communication with other EDPs or the center. Consequently, the dynamics of the caching state can be characterized as follows:

$$\begin{aligned}\mathbf{d}q_{i,k}(t) = Q_k[&-w_1x_{i,k}(t) - w_2\Pi_{i,k}(t) \\ &+w_3\xi^{L_{i,k}(t)}]\mathbf{d}t + \varrho_q\mathbf{d}W_i(t),\end{aligned} \quad (4)$$

where w_1, w_2, w_3, and ϱ_q are positive constant coefficients to adjust the proportion of different factors, and $\xi \in (0,1)$ is a pre-fixed parameter. Similar to Eq. (1), $W_i(t)$ is also a standard Brownian motion. The first term reflects the decrementing rate of the remaining space, which is controlled by the determined caching strategy. The terms $\Pi_{i,k}(t)$ and $\xi^{L_{i,k}(t)}$ signify the incrementing rate of the remaining space, arising from the discarding part based on the dynamic demands of requesters. More concretely, EDP i will discard content k when it is scarcely requested by requesters, i.e., the fewer the requests for content k, the faster it is removed from the caching storage. Meanwhile, EDP i will also appropriately discard content k when it is not urgently needed by requesters, which is a decreasing function of $L_{i,k}(t)$. Here, ξ regulates the steepness of the function. That is, the more urgent the requests for content k, the faster it is added into the caching storage.

During the content trading phase, each EDP sells its own contents to the corresponding requesters. Let $p_{i,k}(t)$ denote the unit price customized by EDP i for selling content k at time t. It is worth noting that EDP i may not cache the whole data of content k in accordance with its predetermined caching strategy $x_{i,k}(t)$. When asking for uncached data from the center or other EDPs, each EDP will give priority to adjacent EDPs for the sake of reducing the transmission delay. Nevertheless, without a proper economic incentive, each EDP is generally unwilling to share its cached data [32]. Hence, we adopt a common usage-based pricing scheme [33], [34], where each EDP needs to pay a uniform unit price (denoted by $\bar{p}_k$) for obtaining content k from other peer EDPs.

C. Model Summary and Problem Statement

Now, we summarize the key characteristics of the above system. Firstly, it is a joint edge caching and pricing framework, where EDPs can determine their own caching strategies by themselves, share their own contents with nearby peers, and trade their contents with requesters. Secondly, we take the dynamics of network conditions and caching storages into consideration, which involves several realistic issues such as the heterogeneous demands of content popularity and timeliness, the real-time trading price, as well as the paid sharing among EDPs. Thirdly, there exists an intertwined game among multiple EDPs under time-varying requests, and the decisions of different EDPs will interact with each other. In this paper, we take an interest in dealing with these problems:

1) How does each EDP customize its own caching strategy and trading price with incomplete information?
2) How do the strategies of different EDPs affect each other?
3) How does one achieve the system/game equilibrium state?

These problems encompass the significant technical and economic interactions among diverse EDPs in the large-scale dynamic MEC system. For ease of reference, we summarize the commonly used notations in Table I.

III. Game Formulation

A. Utility Function Design

The goal of each EDP i is to maximize its own utility (i.e., net profit) by determining a suitable caching strategy $x_{i,k}(t)$ during a finite time horizon T. EDP i will confront with three cases when responding to requesters, i.e.,

- Case 1: EDP i has already cached enough content k for requesters. Owing to the continuity of caching states, some contents might not be wholly cached, i.e., only a relatively small portion α (e.g., $\alpha = 20\%$) is not stored. In this case, the content can basically meet the use of requesters.
- Case 2: EDP i does not cache enough content k but some adjacent EDPs have cached the content. Then, EDP i will associate with the other EDP that provides enough content k and will pay the corresponding reward.
- Case 3: Both EDP i and other EDPs do not cache enough content k. Consequently, the EDP needs to download the uncached portion of content k from the cloud center.

For ease of exposition, let $\mathbb{P}^1(q_{i,k}(t))$, $\mathbb{P}^2(q_{i,k}(t), q_{-i,k}(t))$, and $\mathbb{P}^3(q_{i,k}(t), q_{-i,k}(t))$ describe the occurrence probability of the above three cases, respectively. Here, $q_{-i,k}(t)$ denotes the caching state of the other possible EDP in case 2. More specifically, we can define these probabilities as follows:

$$\mathbb{P}^1(q_{i,k}(t)) = f(\alpha \cdot Q_k - q_{i,k}(t));$$
$$\mathbb{P}^2(q_{i,k}(t), q_{-i,k}(t)) = f(q_{i,k}(t) - \alpha \cdot Q_k) \cdot f(\alpha \cdot Q_k - q_{-i,k}(t));$$
$$\mathbb{P}^3(q_{i,k}(t), q_{-i,k}(t)) = f(q_{i,k}(t) - \alpha \cdot Q_k) \cdot f(q_{-i,k}(t) - \alpha \cdot Q_k),$$

where $f(x) = 1/(1 + e^{-2lx})$, $l > 0$ is a smooth approximation of the heaviside step function. For instance, in case 1, $\mathbb{P}^1$ approaches 1 when $q_{i,k}(t)$ is less than $20\% \cdot Q_k$.

Then, we define the utilities of EDPs based on the above cases, the network model, and the edge caching model. The utility of each EDP refers to the income paid by requesters and the sharing benefit from other EDPs minus total cost, i.e.,

1) **Trading income** $\Phi^1_{i,k}(t)$: Considering the supply-demand relationship in the trading market, we define the dynamic price for each EDP i selling content k as follows:

$$p_{i,k}(t) = \begin{cases} \hat{p}, & M = 1 \\ \hat{p} - \dfrac{\eta_1 \sum_{i'=1, i' \neq i}^{M} Q_k \cdot x_{i',k}(t)}{M-1}, & M \geq 2 \end{cases} \tag{5}$$

where $\hat{p}$ represents the maximum price per unit data that EDPs can charge, and η_1 is used to convert the average content supply into monetary value. According to the principle of supply and demand, when multiple EDPs offer the same content in the trading market, the price of the content tends to decrease as EDPs try to gain a competitive advantage. Based on Eq. (5), the trading income of EDP i can be expressed as:

$$\begin{aligned} \Phi^1_{i,k}(t) &= I_{i,k}(t) p_{i,k} \mathbb{P}^1 (Q_k - q_{i,k}(t)) \\ &+ I_{i,k}(t) p_{i,k} \mathbb{P}^2 (Q_k - q_{-i,k}(t)) + I_{i,k}(t) p_{i,k} \mathbb{P}^3 Q_k. \end{aligned} \tag{6}$$

2) **Sharing benefit** $\Phi^2_{i,k}(t)$: The sharing benefit is the monetary benefit formed when EDP i shares its contents with other EDPs. We use $\mathcal{M}_{i,k}(t) \in \mathcal{M}$ to denote the set of EDPs who request content k from EDP i at time t. Then, the sharing benefit can be expressed as follows:

$$\Phi^2_{i,k}(t) = \sum\nolimits_{i' \in \mathcal{M}_{i,k}(t)} \bar{p}_k (q_{i',k}(t) - q_{i,k}(t)). \tag{7}$$

3) **Content placement cost** $C^1_{i,k}(t)$: Placing and storing cached contents will consume some resources (such as processing capacity, computation time, etc.), which can be characterized by a quadratic function like in [30], i.e.,

$$C^1_{i,k}(t) = w_4 x_{i,k}(t) + w_5 x^2_{i,k}(t). \tag{8}$$

Here, w_4 and w_5 are adjustment coefficients.

4) **Staleness cost** $C^2_{i,k}(t)$: The staleness cost is defined as a penalty function of the total request service delay. Since the delay directly affects QoE for requesters, the function needs to be non-decreasing and nonnegative to measure the impact of staleness. For simplicity, we adopt a linear penalty function to describe the staleness cost, shown as follows:

$$\begin{aligned} C^2_{i,k}(t) = \eta_2 \Big\{ & \frac{Q_k x_{i,k}(t)}{H_c} + \sum\nolimits_{j \in I_{i,k}(t)} \Big[\mathbb{P}^1 \frac{Q_k - q_{i,k}(t)}{H_{i,j}(t)} \\ & + \mathbb{P}^2 \frac{Q_k - q_{-i,k}(t)}{H_{i,j}(t)} + \mathbb{P}^3 \Big(\frac{q_{i,k}(t)}{H_c} + \frac{Q_k}{H_{i,j}(t)} \Big) \Big] \Big\}, \end{aligned} \tag{9}$$

where η_2 is utilized to convert the total request service delay into the staleness cost, and H_c denotes the transmission rate between the center and any EDP. The first term is generated by the process in which EDP i downloads content k from the center according to the determined caching strategy. The second term corresponds to case 1 when EDP i can directly transmit content k to requesters. The third term denotes the delay when the EDP purchases the content from an adjacent

EDP and transfers it to requesters. Here, we omit the transmission time between EDP i and its adjacent EDPs because it is much smaller than the communication delay between EDP i and the center/requesters. The fourth term corresponds to case 3 when EDP i will download the uncached content from the data center and transmit the whole content k to requesters. After accumulating the delay of all corresponding requesters, the staleness cost of EDP i can be acquired.

5) **Sharing cost** $C^3_{i,k}(t)$: In case 2, EDP i provides a remuneration to a suitable adjacent EDP who is willing to share content k, which will yield the sharing cost and can be expressed by $C^3_{i,k}(t)=\mathbb{P}^2\bar{p}_k(q_{i,k}(t)-q_{-i,k}(t))$.

Based on the above definitions, the utility function of EDP i with respect to content k in time t can be represented as:

$$\mathbf{U}_{i,k}(t)=\Phi^1_{i,k}(t)+\Phi^2_{i,k}(t)-C^1_{i,k}(t)-C^2_{i,k}(t)-C^3_{i,k}(t). \quad (10)$$

It is noteworthy that the net profit of each EDP not only depends on its own strategy but also is affected by other EDPs' strategies. However, each EDP is unaware of the strategies/states of other EDPs when making caching decisions.

B. Game Formulation

Based on the above models, we formulate the competitive content placement problem among large-scale EDPs as a stochastic differential game with incomplete information in a finite time horizon, presented as follows:

- Players: A set $\mathcal{M}=\{1,\cdots,i,\cdots,M\}$ of EDPs.
- States: Let the 2-tuple $S_{i,k}(t)=\langle\{h_{i,j}(t)|j\in\mathcal{J}_i(t)\},q_{i,k}(t)\rangle$ denote the state of EDP i at time t with respect to a given content k. Then, the state space is $\{S_{i,k}(t)|i\in\mathcal{M},t\in[0,T]\}$.
- Strategies: A set $\{x_{i,k}(t)\}$ of all possible caching controls.
- Accumulative utility: The accumulative utility of player i with regard to content k over the finite time horizon t can be defined by $\mathcal{U}_{i,k}(t)=\mathbb{E}[\int_0^t \mathbf{U}_{i,k}(t')\mathbf{d}t']$.

In the above stochastic differential game, the optimization problem of each EDP can be formulated as follows:

$$\max_{x_{i,k}(t),t:0\to T} \quad \mathcal{U}_{i,k}(x_{i,k},S_{i,k},\boldsymbol{S}_{-i,k}), \quad (11)$$

$$\begin{aligned} s.t., \quad \mathbf{d}h_{i,j}(t) &= \frac{1}{2}\varsigma_h(\upsilon_h-h_{i,j}(t))\mathbf{d}t+\varrho_h\mathbf{d}W_{i,j}(t),\\ \mathbf{d}q_{i,k}(t) &= Q_k[-w_1x_{i,k}(t)-w_2\Pi_{i,k}(t)\\ &\quad +w_3\xi^{L_{i,k}(t)}]\mathbf{d}t+\varrho_q\mathbf{d}W_i(t), \quad (12)\end{aligned}$$

where $\boldsymbol{S}_{-i,k}=\{s_{i',k}|i'\neq i,i'\in\mathcal{M}\}$ means the states of other $M-1$ EDPs. The solution of the optimization problem needs to achieve the Nash equilibrium, which is defined below:

Definition 3 (***Nash Equilibrium, NE*** [35])**.** The set of optimal caching strategies $\{\boldsymbol{x}^*_i(t)|i\in\mathcal{M},t\in[0,T]\}$ constitutes a Nash equilibrium if and only if the following inequality holds:

$$\mathcal{U}_{i,k}(x^*_{i,k}(t),S^*_{i,k}(t),\boldsymbol{S}^*_{-i,k}(t))\geq\mathcal{U}_{i,k}(x_{i,k}(t),S^*_{i,k}(t),\boldsymbol{S}^*_{-i,k}(t)),$$

where $S^*_{i,k}(t)$ and $\boldsymbol{S}^*_{-i,k}(t)$ denote the corresponding optimal states under the optimal caching strategies. **Def. 3** can guarantee that no participant can improve its own utility by unilaterally deviating from its optimal strategy.

According to the theory of dynamic programming, the NE of the game can be achieved by solving a series of Hamilton-Jacobi-Bellman (HJB) equations to characterize the optimal strategies for players. The HJB equation is a partial differential equation that describes the value function associated with the game, where the value function is the optimization objective, i.e., $\mathcal{V}_{i,k}(t,S_{i,k},\boldsymbol{S}_{-i,k})=\max_{x_{i,k}(t)}\mathcal{U}_{i,k}(x_{i,k},S_{i,k},\boldsymbol{S}_{-i,k})$. For each EDP $i\in\mathcal{M}$, the HJB equation can be presented as:

$$\max_{x_{i,k}(t)}\left[\mathcal{O}\mathcal{V}_{i,k}(t)+\mathbf{U}_{i,k}(t)\right]+\partial_t\mathcal{V}_{i,k}(t)=0, \quad (13)$$

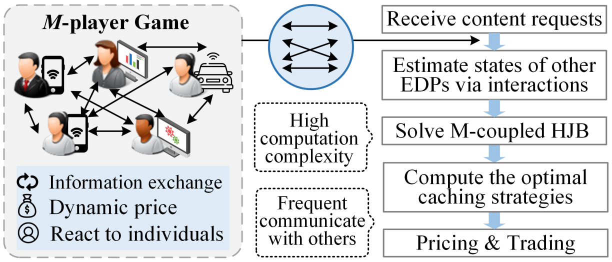
where $\mathcal{O}$ represents the partial differential operator. The optimal caching strategy of each EDP can be derived by solving the M equations simultaneously. Unfortunately, there exists a complicated coupling of the M equations, and it is arduous to solve them. This is because each EDP needs the caching strategies and state information of all other EDPs, leading to a high computational burden and significant communication overhead. Moreover, the content trading process involves a dynamic pricing policy and a peer-to-peer content sharing policy, which also results in the unknown mutual influence among EDPs' decisions. Therefore, it is unrealistic to obtain the solution of equations directly when the number of EDPs is enormous. On the other hand, when each EDP wishes to customize its own strategy by itself in a decentralized manner, the MEC system finds it harder to achieve an NE in such an incomplete information scenario. To solve these challenges, we further propose a new game framework below.

IV. Framework Design

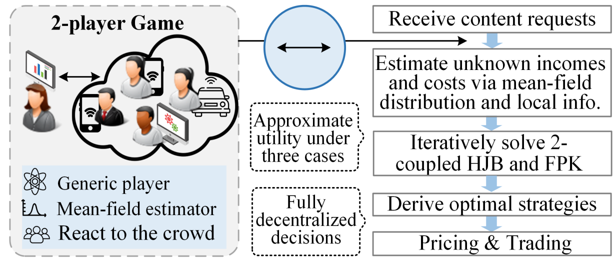
In this section, we focus on designing a joint game framework, called MFG-CP. We first introduce the basic idea of MFG-CP. Next, we derive the optimal caching strategy for each EDP and present the detailed framework descriptions. Finally, the NE and complexity of MFG-CP are analyzed.

A. Basic Idea

We propose a joint game framework for caching and pricing (i.e., MFG-CP), in which each EDP can directly customize its optimal caching strategy and pricing policy without knowing other EDPs' strategies, states, trading prices, and sharing effects. The basic idea of MFG-CP is to approximate the collective impact of large-scale EDPs on content caching, trading, and sharing without considerable interactions, and then iteratively solve multiple coupled equations to determine the optimal caching strategy for each EDP. More explicitly, we first transform the original stochastic differential game to an MFG, where the multi-player interactions can be regarded as a generic-player representation based on a statistical distribution. Then, we design a mean-field estimator to approximate the cumulative impact of all EDPs on caching, trading, and sharing, in which the Fokker-Planck-Kolmogorov (FPK) equation is constructed to display the evolution of mean-field distribution. Here, we skillfully estimate the unknown incomes and costs of each EDP without the need for frequent information exchange. Next, we design a generic player to adjust its caching strategy based on the HJB equation and the feedback from the mean-field estimator. Finally, in order to acquire the NE of the game for each EDP in a distributed scenario, we design an iterative best response learning algorithm to solve the coupled HJB and FPK equations. In this way, the optimal caching strategy



(a) Framework under the stochastic differential game



(b) MFG-CP under the mean-field game

Fig. 2. Comparisons under different game frameworks.

can be determined during the time horizon using only local information and a mean-field distribution.

As illustrated in Fig. 2, we compare MFG-CP with the original game. Clearly, the number of partial differential equations used for deriving the optimal caching strategies in the MFG-CP is reduced from $M \times K$ to $2 \times K$. In general, according to the Zipf law, the request probability of most contents remains considerably small, even when K is large. Hence, the number of equations that need to be solved is less than $2 \times K$, implying that the computing time required to derive the optimal strategies will not incur excessive complexity.

B. Determining the Optimal Caching Strategy

Firstly, we transform the original game into a novel MFG and design two modules: a generic player who takes rational actions and a mean-field estimator that represents the collective actions of all other players. At the beginning, each generic EDP formulates a strategy set comprising all feasible states to optimize its utility. Since all EDPs are symmetrical in possible states and are equally rational, they share a common strategy set. Subsequently, by approximating the Probability Density Function (PDF) of states, the mean-field estimator calculates the cumulative impact of all EDPs on content caching, trading, and sharing for the generic EDP. In this way, the generic EDP adjusts its strategy set based on the feedback from the mean-field estimator, and the mean-field estimator recalculates the impact according to the updated strategy set. This iterative process continues until an NE is achieved.

Then, we introduce the two modules: the mean-field estimator and the generic player, respectively.

(1) ***Mean-field estimator:*** The above MFG model possesses four characteristics: the rationality of EDPs; a large number of EDPs; the interchangeability of the states of EDPs; and the interactions among EDPs via the mean-field estimator. The first feature shows that each EDP makes decisions by itself. The second feature shows the large-scale MEC system possesses a sufficient number of EDPs, so that the role of behavior decision for each EDP in the group will become smaller. In other words, the state exchange among EDPs will not alter the outcome of the game, which corresponds to the third feature. The presence of the fourth feature arises from the fact that each EDP interacts with the mean-field estimator, rather than an individual within the environment.

Building upon the aforementioned characteristics, the caching strategy of each EDP is only determined by its own current state and the mean-field estimation. That is, any permutation of the indexes of EDPs will not change the determination of the optimal strategy. Consequently, we can concentrate on a generic EDP by removing the index i, and the EDP interacts with the mean-field estimator instead of communicating with all the other EDPs separately.

Given the state $S_k(t)$, the mean-field estimator needs to estimate the statistical distribution of this state at time t, and the probability density of EDPs in a specific state with respect to content k is expressed as follows:

$$\lambda(S_k(t)) = \lim_{M\to\infty} \Lambda(S_k(t)) = \lim_{M\to\infty} \frac{1}{M} \sum\nolimits_{i=1}^{M} \mathbb{1}_{\{S_{i,k}(t)=S_k(t)\}}, \quad (14)$$

where $\Lambda(S_k(t))$ means the proportion of EDPs in the state $S_k(t)$, and the indicator function $\mathbb{1}_{\{\cdot\}}$ returns one if the given condition is satisfied and zero otherwise. Since $\lambda(S_k(t))$ is a continuum PDF, there is $\int_h \int_q \lambda(S_k(t))\mathbf{d}h\mathbf{d}q = 1$. When the number of EDPs approaches infinity (i.e., $M \to \infty$), the proportion $\Lambda(S_k(t))$ converges to a mean-field PDF $\lambda(S_k(t))$, which characterizes the state evolution of the EDPs over time. When all EDPs in the MEC system implement the derived optimal strategies (i.e., **Theorem 1**), the evolution of the mean-field distribution under the dynamics of the channel state $h(t)$ and the remaining caching state $q_k(t)$ can be described by leveraging the FPK equation as follows:

$$\begin{aligned} &\partial_t \lambda(S_k(t)) + \frac{1}{2}\varsigma_h(\upsilon_h - h(t))\partial_h \lambda(S_k(t)) \\ &+ Q_k[-w_1 x_k(t) - w_2 \Pi_k(t) + w_3 \xi^{L_k(t)}]\partial_q \lambda(S_k(t)) \\ &- \frac{1}{2}\varrho_h^2 \partial_{hh}^2 \lambda(S_k(t)) - \frac{1}{2}\varrho_q^2 \partial_{qq}^2 \lambda(S_k(t)) = 0. \end{aligned} \quad (15)$$

Based on the mean-field distribution, we can approximate the unknown incomes and costs which involve the strategies and states of other EDPs. Firstly, we need to determine the dynamic price $p_k(t)$ with the help of the mean-field PDF $\lambda(S_k(t))$ and the MFG-based optimal strategy $x_k^*(S_k(t))$. According to Eq. (5), the price is associated with other EDPs' caching strategies, which can be rewritten as:

$$\begin{aligned} p_{i,k}(t) &= \hat{p} - \frac{\eta_1 M Q_k}{M-1} \int_h \int_q \Lambda(S_k(t)) x_k^*(S_k(t)) \mathbf{d}h\mathbf{d}q \\ &+ \frac{\eta_1 Q_k}{M-1} x_{i,k}(S_k(t)) \Rightarrow p_k(t) = \lim_{i\to\infty} p_{i,k}(t), \end{aligned} \quad (16)$$

$$\Rightarrow p_k(t) \approx \hat{p} - \eta_1 Q_k \int_h \int_q \lambda(S_k(t)) x_k^*(S_k(t)) \mathbf{d}h\mathbf{d}q. \quad (17)$$

Here, Eq. (17) holds due to the fact that $\lim_{i\to\infty} M/(M-1) = 1$ and $\lim_{i\to\infty} \frac{\eta_1}{M-1} x_{i,k}(S_k(t)) = 0$. In addition to the dynamic price, the trading income also needs to know $q_{-i,k}(t)$, which can be approximated by the average state of all other EDPs. More specifically, we define $\bar{q}_{-,k}(t)$ to represent the mean remaining caching storage space of other EDPs. Given the mean-field distribution, the average caching state with regard to $\lambda(S_k(t))$ can be expressed as:

$$\bar{q}_{-,k}(t) \approx \int_h \int_q q_k(t)\lambda(S_k(t))\mathbf{d}h\mathbf{d}q_k. \tag{18}$$

Then, we estimate the sharing benefit to analyze the influence of paid content sharing. We assume that the center will randomly assign a suitable EDP to respond to the corresponding EDP's request. Hence, we consider the average sharing benefit for sharing content k, i.e., the monetary benefit acquired by each EDP who has cached enough content k. At time t, the number of EDPs who have the qualification to share their content k can be approximated as $M_k(t) = \sum_{i=1}^{M} \mathbb{1}_{q_{i,k}(t) \leq \alpha \cdot Q_k}$, which can be publicized by the center. Meanwhile, the center also observes the number of EDPs who have encountered case 3, denoted by $M_k'(t)$. In addition, the average transmission size $\overline{\Delta q}(t)$ between EDPs can be calculated as follows:

$$\overline{\Delta q}(t) \approx \left| \iint_{q_k \leq \alpha \cdot Q_k} q_k(t)\lambda(S_k(t))\mathbf{d}h\mathbf{d}q_k - \iint_{q_k > \alpha \cdot Q_k} q_k(t)\lambda(S_k(t))\mathbf{d}h\mathbf{d}q_k \right|.$$

From a holistic perspective, the average sharing benefit at time t can be expressed as: $\bar{\Phi}_k^2(t) = \bar{p}_k \overline{\Delta q}(t)(\frac{M - M_k'(t)}{M_k(t)} - 1)$.

According to the above approximation, the generic EDP can further optimize its utility by calling the mean-field estimator and can determine the optimal caching strategy under the current state information. The process of the strategy determination is elucidated in the following module.

(2) ***Generic player:*** With the establishment of the mean-field estimator, the optimization problem of each EDP can be converted into a generic problem, which only involves the local state information and the mean-field distribution, i.e.,

$$\max_{x_k(0 \to T)} \quad \mathcal{U}_k(x_k, S_k, \lambda). \tag{19}$$

It is crucial to highlight that the utility function in Eq. (19) is exactly consistent with Eq. (11) after removing the EDP index i. More importantly, we just employ the mean-field estimator to quickly obtain $p_k(t), \bar{q}_{-,k}(t)$, and $\bar{\Phi}_k^2(t)$, and then we directly substitute them for the trading income, sharing benefit, staleness cost, and sharing cost. As a result, the value objective function $\mathcal{V}_k(t, S_k, \lambda)$ only relies on the individual caching strategy, the current state, and the mean-field distribution. To address the optimization problem, the HJB equation (i.e., Eq. (13)) can be rewritten as:

$$\begin{aligned} &\max_{x_k(t)} \Bigg[\frac{1}{2}\varsigma_h(\upsilon_h - h(t))\partial_h \mathcal{V}_k(t) + \frac{1}{2}\varrho_h^2 \partial_{hh}^2 \mathcal{V}_k(t) \\ &+ Q_k[-w_1 x_k(t) - w_2 \Pi_k(t) + w_3 \xi^{L_k(t)}]\partial_q \mathcal{V}_k(t) \\ &+ \frac{1}{2}\varrho_q^2 \partial_{qq}^2 \mathcal{V}_k(t) + \mathbf{U}_k(t, x_k, S_k, \lambda) \Bigg] + \partial_t \mathcal{V}_k(t) = 0. \end{aligned} \tag{20}$$

Based on the value function $\mathcal{V}_k(t)$ acquired by solving Eq. (20), the optimal caching strategy can be derived by **Thm. 1**.

Theorem 1. *In the MFG-CP framework, the optimal caching strategy can be determined by*

$$x_k^*(t) = \left[-(\frac{w_4}{2w_5} + \frac{\eta Q_k}{2H_c w_5} + \frac{Q_k w_1 \partial_q \mathcal{V}_k(t)}{2w_5}) \right]^+, \tag{21}$$

where the function $[x]^+$ *is defined as:* $[x]^+ = 1$ *when there is* $x > 1$, $[x]^+ = 0$ *when there is* $x < 0$, *and* $[x]^+ = x$ *otherwise.*

Proof. According to the HJB equation in Eq. (20), the optimal strategy of the MFG game is the argument of the supremum term, i.e., $x_k^*(t) = \max(\Omega(t, x_k, S_k, \lambda))$, where $\Omega(t, x_k, S_k, \lambda) = \frac{1}{2}\varsigma_h(\upsilon_h - h(t))\partial_h \mathcal{V}_k(t) + \frac{1}{2}\varrho_h^2 \partial_{hh}^2 \mathcal{V}_k(t) + Q_k[-w_1 x_k(t) - w_2 \Pi_k(t) + w_3 \xi^{L_k(t)}]\partial_q \mathcal{V}_k(t) + \frac{1}{2}\varrho_q^2 \partial_{qq}^2 \mathcal{V}_k(t) + \mathbf{U}_k(t, x_k, S_k, \lambda)$. The maximum term is a convex function of $x_k(t)$ for all time t, because the second order derivative is lower than zero. Therefore, we can employ the Karush-Kuhn-Tucker conditions to find the critical point, i.e., computing the first order derivative of $\Omega(t, x_k, S_k, \lambda)$ as follows:

$$\frac{\partial \Omega(t, x_k, S_k, \lambda)}{\partial x_k(t)} = -Q_k w_1 \partial_q \mathcal{V}_k(t) - (w_4 + 2w_5 x_k(t)) - \frac{\eta_2 Q_k}{H_c}.$$

We let the above derivative equal to zero, and then the closed-form expression of the optimal caching strategy can be derived as presented in Eq. (21).

Besides, it is noteworthy that $x_k^*(t)$ is associated with $\mathcal{V}_k(t)$, which is the solution of Eq. (20). By substituting $x_k^*(t)$ back into Eq. (20), the final version of the HJB equation without the variable $x_k(t)$ can be directly obtained. □

The solution under the MFG-CP framework outlined above is nearly equivalent to that of the stochastic differential game when dealing with a large number of players, ensuring consistency in the optimization problem. Additionally, the mean-field approximation method designed in our framework exhibits lower complexity compared to the original game. As can be seen from Eqs. (15) and (20), these two modules interact with each other, forming an integral part of the overall MFG-CP framework. More specifically, the solution of the HJB equation has a great impact on the FPK equation to update the mean field. Conversely, the solution of the FPK equation is also required by the HJB equation to estimate the generic player's utility and update strategies. These interlinked modules comprehensively construct the core structure of MFG-CP, which will be elaborated further below.

C. Framework Description

The detailed game framework for mobile edge caching and pricing in a continuous optimization epoch is presented in **Alg. 1**, in which the optimal caching strategy can be determined by the iterative learning process according to **Alg. 2**. First of all, the state of each EDP and the data size of each content are initialized (Line 1). By utilizing parallel processing, each EDP takes the finite time horizon T as an optimization epoch (Lines 2-14). Specifically, each EDP initially records the requirements of requesters $I_{i,k}(t)$ and determines the content set $\mathcal{K}'$ that needs to be cached (Lines 4-5). Here, we assume that the change in requesters' demands occurs at a relatively slow rate compared to the time scale of the optimization epoch. Therefore, some contents need not be considered in the following optimization. Subsequently, each EDP conducts the edge caching and content trading for each content $k \in \mathcal{K}'$ (Lines 6-14). Based on Defs. 1 and 2, the content popularity and content timeliness can be calculated (Line 8). Then, EDP

Algorithm 1: The Proposed Framework (MFG-CP)

```
input : The parameters w_1 ~ w_5, η_1, η_2, ς_h, υ_h, ϱ_h, ϱ_q, and
        ξ; the maximum epoch number σ_max;
1  Initialize: the state S(0); the data size of each content Q_k;
2  for each EDP i = 1, 2, ···, M in parallel do
3      while each optimization epoch σ ≤ σ_max do
4          Record requests {I_{i,k}(t) | k ∈ K, t ∈ [σT, (σ+1)T)};
5          Determine the content set K' = {k | q_{i,k}(t) < Q_k
            and Σ_{σT}^{(σ+1)T} |I_{i,k}(t)| > 0} that needs to be cached;
6          for each content k ∈ K' do
7              // Mobile Edge Caching;
8              Compute content popularity Π_{i,k}(t) and con-
                tent timeliness L_{i,k}(t) for all contents;
9              Call for Alg. 2 to obtain the optimal caching
                strategy x*_{i,k}(t) as its best response;
10             Update the current state based on x*_{i,k}(t);
11             // Pricing and Trading;
12             Case 1: sell the content at a unit price p_{i,k}(t);
13             Case 2: buy the uncached content from an
                adjacent EDP at p̄_k and sell it to requesters;
14             Case 3: download the uncached content from
                the center and sell it to requesters;
```

Algorithm 2: Iterative Best Response Learning Scheme

```
input : The iterative number ψ; iterative threshold ψ_th;
output: The optimal caching strategy;
1  Initialize: ψ = 0; x_k^ψ(0); V_k^ψ((σ+1)T); λ;
2  while ψ < ψ_th do
3      ψ = ψ + 1;
4      Solve the HJB equation using x_k^{ψ-1}(t) and λ;
5      Update the strategy vector x_k^ψ(t) based on Eq. (21);
6      if |x_k^ψ(t) − x_k^{ψ-1}(t)| < a preset threshold then
7          Break the iterative learning process;
8      Solve the FPK equation using x_k^ψ(t) to update λ;
9      Update the mean-field estimator: p_k(t), q̄_{−,k}(t), Δq̄(t);
10     Update the utility function according to Eq. (10);
11 return x_k^ψ(t) for t ∈ [σT, (σ+1)T).
```

i invokes **Alg. 2** to give the best response, i.e., customizing its optimal caching strategy (Lines 9-10). After that, the trading process between the EDP and the corresponding requesters will be carried out, where the EDP needs to take different actions under various cases (Lines 11-14).

Alg. 2 describes the iterative process for learning the best response for each EDP. When $\psi<\psi_{th}$ or $|x_k^{\psi}(t)-x_k^{\psi-1}(t)|$ is greater than a preset threshold, we iteratively solve the coupled HJB equation and FPK equation: the backward HJB models the induction process of the optimization of each individual, while the forward FPK models the evolution of the mean-field as a whole. According to Eqs. (20) and (21), we solve the HJB equation to update the strategy $x_k^{\psi}(t)$ (Lines 4-5). Then, we compute the FPK equation to update the mean-field estimator based on Eq. (15) and the current strategy. Meanwhile, the utility function can be updated according to Eq. (10), which will be used for solving the HJB equation in the next iteration.

Remark: For each optimization epoch, the computational complexity of **Alg. 1** is $O(K\psi_{th})$, where ψ_{th} is the iterative threshold. It is worth noting that the complexity of MFG-CP is significantly lower than the complexity $O(MK\psi_{th})$ of a general framework that requires frequent information exchange among all EDPs. Consequently, the convergence time of MFG-CP does not increase with M, effectively addressing the scaling problem of the original game. Moreover, MFG-CP can be easily extended to the scenario whereby the caching capacity of each EDP is less than a fixed threshold. In fact, this further optimization can be seen as a knapsack problem, in which the weight and value of each content are considered. Based on the solution of MFG-CP, the final caching strategy will be further derived by solving the knapsack problem.

D. Theoretical Analysis

The optimal caching policy can be determined by the mutual iterative calculation of HJB-FPK equations over the finite time horizon. Hence, we first prove the existence and uniqueness of the HJB equation and the FPK equation, respectively.

Lemma 1. *In the MFG-CP framework, there exists the unique value function $\mathcal{V}_k(t)$ of the HJB equation.*

Proof. When the two conditions hold, the value function is the unique solution of the HJB equation [36]: (i) the caching strategy space is a compact subset of $\mathbb{R}$; (ii) the drift term of the state dynamics and the utility function are bounded and Lipschitz continuous. The first condition is clearly satisfied since the caching strategy space is confined to $[0,1]$. For the second condition, we need to analyze the drift terms and the utility function, respectively. Based on Eqs. (1) and (4), the drift terms are listed as follows:

$$DF_1(t,h_k,x_k)=\varsigma_h(\upsilon_h-h_k(t))/2,$$

$$DF_2(t,q_k,x_k)=Q_k[-w_1x_k(t)-w_2\Pi_k(t)+w_3\xi^{L_k(t)}].$$

We observe that DF_1 and DF_2 are bounded because $h_k(t)$ is bounded and $\xi\in(0,1)$. Therefore, the drift term of system dynamics $|DF(t,S_k,x_k)|=\sqrt{DF_1^2+DF_2^2}$ is bounded. Due to $\Delta DF_1=|DF_1(t,h_k,x_k)-DF_1(t,h_k',x_k)|=\varsigma_h(h_k'-h_k)/2$ and $\Delta DF_2=|DF_2(t,q_k,x_k)-DF_2(t,q_k',x_k)|=0$, there is:

$$|DF(t,S_k,x_k)-DF(t,S_k',x_k)|=\sqrt{(\Delta DF_1)^2+(\Delta DF_2)^2}$$
$$\le|\varsigma_h/2|\sqrt{|h_k'-h_k|^2+|q_k'-q_k|^2}. \tag{22}$$

Therefore, the drift term is Lipschitz continuous.

For the utility function shown in Eq. (10), we can easily find that $\mathbf{U}_k(t)$ is bounded, and the partial derivative of $\mathbf{U}_k(t)$ is:

$$\partial_S\mathbf{U}_k(t)=\partial_S\Phi_k^1(t)+\partial_S\Phi_k^2(t)-\partial_S C_k^2(t)-\partial_S C_k^3(t). \tag{23}$$

Based on Eq. (23), we further deduce the derivative of $\mathbf{U}_k(t)$ with regard to $q_k(t)$, which can be calculated as:

$$\partial_q\mathbf{U}_k(t)\le I_k(t)p_k(t)\big[\partial_q\mathbb{P}^1(Q_k-q_k(t))+\partial_q\mathbb{P}^2(Q_k-q_{-,k}(t))$$
$$-\mathbb{P}^1+Q_k\partial_q\mathbb{P}^3\big]-\sum\nolimits_{j\in I_k(t)}\big[\partial_q\mathbb{P}^1\frac{Q_k-q_k(t)}{H_j(t)}-\frac{\mathbb{P}^1}{H_j(t)}$$
$$\partial_q\mathbb{P}^2\frac{Q_k-q_{-,k}(t)}{H_j(t)}+\partial_q\mathbb{P}^3(\frac{q_k(t)}{H_c}+\frac{Q_k}{H_j(t)})+\frac{\mathbb{P}^3}{H_c}\big]$$

$$-\partial_q \mathbb{P}^2 \bar{p}_k (q_k(t) - q_{-,k}(t)) - \mathbb{P}^2 \bar{p}_k, \tag{24}$$

where $\partial_q \mathbb{P}^1 = f'(\alpha \cdot Q_k - q_k(t))$, $\partial_q \mathbb{P}^2 = f'(q_k(t) - \alpha \cdot Q_k) \cdot f(\alpha \cdot Q_k - q_{-,k}(t))$, $\partial_q \mathbb{P}^3 = f'(q_k(t) - \alpha \cdot Q_k) \cdot f(q_{-,k}(t) - \alpha \cdot Q_k)$, and $f'(x) = 2le^{-2lx}(1 + e^{-2lx})^{-2}$. Since Eq. (24) is the elementary function of $q_k(t)$ and $q_k(t)$ is bounded, the derivative $\partial_q \mathbf{U}_k(t)$ is also bounded. Similarly, we can verify that the partial derivative with regard to $h_k(t)$ is bounded. Therefore, the utility function is bounded and Lipschitz continuous. In other words, the second condition is also satisfied. □

Lemma 2. *In the MFG-CP framework, there exists the unique mean-field distribution $\lambda(S_k(t))$ of the FPK equation.*

Proof. According to existing works [30], [36], we use the following form to describe a parabolic partial differential equation: $\partial_t \lambda(S_k(t)) + \Theta = d$, where $\Theta = -\sum_{i,j=1}^{k} a_{i,j} \lambda(S_i S_j) + \sum_{i=1}^{k} b_i \lambda(S_i) + c\lambda$. We assume that there is $\lambda \in \Xi$, where Ξ represents the mean-field distribution space. If the following conditions hold: $a_{i,j}, b_i, c \in L^\infty(\Xi)$, $d \in L^2(\Xi)$, and $a_{i,j} = a_{j,i}$, then there exists a weak unique solution λ. After observing the designed FPK equation (i.e., Eq. (15)), we can obtain:

$$c = d = 0; \quad a_{i,j} = \begin{cases} \frac{1}{2}\varrho_h^2 + \frac{1}{2}\varrho_q^2, & i = j = 1, \\ 0, & \text{Otherwise.} \end{cases} \tag{25}$$

Furthermore, we directly get $\| a_{i,j} \|_\infty \leq \frac{1}{2}\varrho_h^2 + \frac{1}{2}\varrho_q^2$, $\| c \|_\infty = 0$, and $\| d \|_2 = 0$. Hence, these conditions $a_{i,j}, c \in L^\infty(\Xi)$, $d \in L^2(\Xi)$, and $a_{i,j} = a_{j,i}$ are satisfied. According to **Lemma 1**, we have demonstrated the boundedness of the drift term b_i, so that $b_i \in L^\infty(\Xi)$ can also be satisfied. Hence, we establish the existence and uniqueness of the solution for the FPK equation, thereby completing the proof of this lemma. □

Building on the aforementioned lemmas, we have established the existence of unique solutions to the HJB and FPK equations in one iteration. Consequently, the optimal caching strategy and mean-field distribution can be acquired through iteratively solving these coupled equations. In the following, we need to guarantee the convergence of the updating process, i.e., proving the uniqueness of the solution pair for the two coupled partial differential equations.

Theorem 2. *In the MFG-CP framework, there exists a unique Nash equilibrium.*

Proof. By using the temporary solutions acquired in the previous iteration as the initial values for the next iteration, each iteration actually involves a contraction mapping $\Xi \times \Xi \to \Xi \times \Xi$. Similar to Lemma 1, we can also prove that the mapping satisfies the assumptions (H1-H5) in [37]. Next, given this contraction mapping, there exists a unique fixed-point based on the fixed-point theorem, i.e., the fixed point can be found by initializing the iterations with an arbitrary point. Owing to the existence of the unique solutions of HJB and FPK equations proven in **Lemma 1** and **Lemma 2**, the iterations for solving Eqs. (15) and (20) are guaranteed to converge to a fixed point $[\mathcal{V}_k^*(t), \lambda^*(S_k(t))]$, which represents the unique solution pair of the HJB-FPK equations. As a result, the existence and uniqueness of the Nash equilibrium can be guaranteed in the MFG-CP framework. The proof is now completed. □

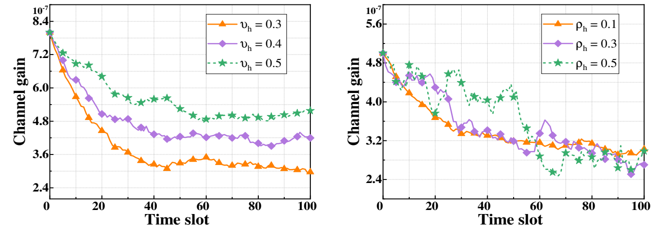


Fig. 3. Channel evolution in the network.

V. Performance Evaluations

In this section, we evaluate the performance of MFG-CP with extensive simulations on real-world datasets. We first present the simulation settings and introduce the compared algorithms, followed by the detailed experimental results.

A. Evaluation Methodology

Simulation Settings: We consider an MEC scenario where EDPs and requesters are randomly distributed within a certain range. In our simulations, we choose a set of $K = 20$ content categories and $M = 300$ EDPs. The number of requests for each category is obtained from real-world YouTube Data [38]. Each trace of the dataset records content id, tags, views, comment count, description, etc. The network parameters are set as follows: $B = 10$MHz, and $\tau = 3$. Meanwhile, the edge caching parameters are set as: $w_1 = 1$, $w_2 = 1/20$, $w_3 = 10$, $w_4 = 2.5 * 10^3$, $w_5 = 0.65 * 10^8$, $\xi = 0.1$, $\varrho_q = 0.1$, $Q_k = 100$MB, $\hat{p} = 5 * 10^{-7}$, $\alpha = 20\%$, and $T = 1$. For simplicity, we assume that all EDPs hold the same transmission power $G = 1$W, and the initial distribution $\lambda(0)$ obeys a normal distribution with mean and standard deviation. Here, the mean value is produced from $[0.5, 0.8]$ and the standard deviation value is chosen from $\{0.05, 0.1\}$. By default, we set $\lambda(0) \sim \mathcal{N}(0.7, 0.1^2)$. Finally, the conversion parameter η_1 changes from $[0.1, 0.4]$, the range of the channel fading coefficient is set as $[1, 10] * 10^{-5}$, and the caching state of any EDP ranges from $[0, 100\text{MB}]$. Moreover, we employ the finite difference method to numerically solve the coupled HJB and FPK equations.

Compared Algorithms: Since existing works do not consider content caching and pricing in incomplete information game scenarios, they cannot be directly applied in our system. Hence, we borrow the basic idea in these works and carefully design four content placement algorithms for comparison: Random Replacement (RR), Most Popular Caching (MPC) [18], MFG [27], and Ultra-Dense Caching Strategy (UDCS) [28]. The RR policy adopts random caching decisions; the MPC method only caches currently most popular contents; the MFG scheme is a downgraded version of MFG-CP, in which the content sharing is not considered; and the UDCS approach takes into account the content overlap and interference, without considering the pricing issue and content sharing.

B. Evaluation Results

1) *Evaluation of Mean-Field Equilibrium:* First, we evaluate the channel evolution in the network to verify the rationality

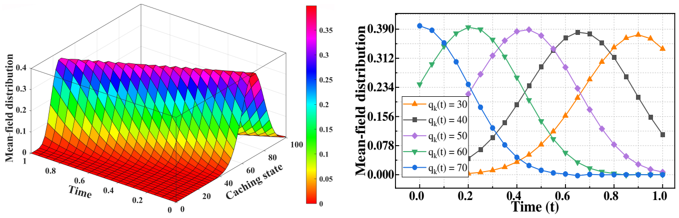


Fig. 4. Evolution of the mean-field distribution at the equilibrium.

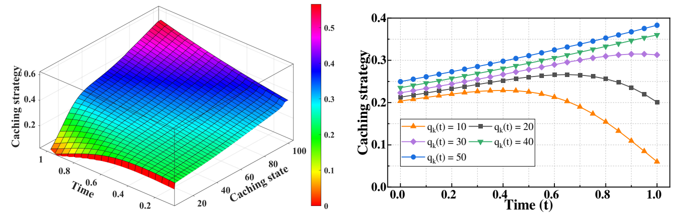


Fig. 5. Evolution of the caching strategy at the equilibrium.

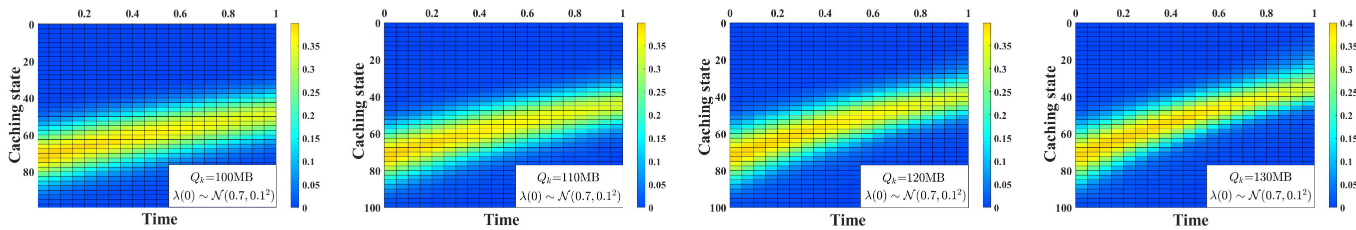


Fig. 6. A heat map description of the mean-field distribution under different Q_k and $\lambda(0) \sim \mathcal{N}(0.7, 0.1^2)$.

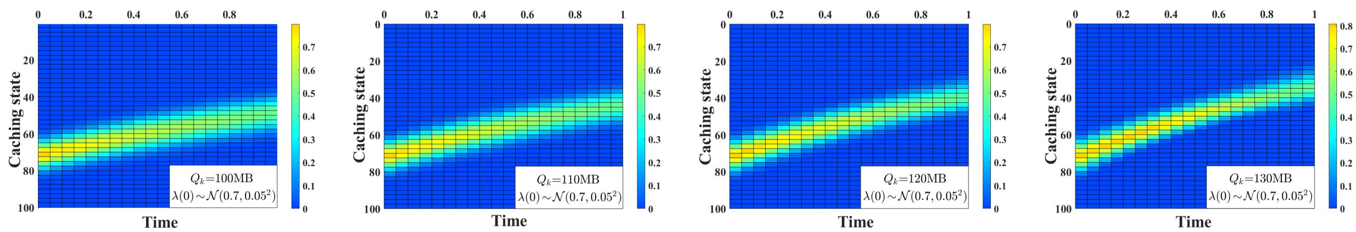


Fig. 7. A heat map description of the mean-field distribution under different Q_k and $\lambda(0) \sim \mathcal{N}(0.7, 0.05^2)$.

of Eq. (1), in which we observe the trend of the channel gain by varying the values of the long-term mean and standard deviation. As illustrated in Fig. 3, the channel gain tends to revert or move back toward a certain mean or equilibrium level under different values of υ_h. Since we set the fixed distance between EDPs and requesters, the evolution of the channel fading coefficient is consistent with Eq. (1). It is important to realize that the fluctuations in this process are caused by the Brownian motion, and we change the values of standard deviation to assess the channel gain. We find that a larger ϱ_h leads to a greater channel deviation trajectory and a less stable channel condition. Consequently, this unstable network transmission environment will certainly have an effect on the performance of EDPs' caching strategies. Therefore, we choose $\varrho_h = 0.1$ in the later simulations.

Then, given an initial caching state, we evaluate the evolution of the mean-field distribution over time at the equilibrium, as shown in Fig. 4. When we fix the time slot, Fig. 4 represents the instantaneous density of EDPs having the remaining caching space $q_k(t)$, and we notice that the size of the remaining caching space will increase first and then decrease. This curve occurs because each EDP dynamically adjusts the trading price and content placement to cache more popular or urgent contents. Subsequently, we maintain the remaining caching space and observe the changing distribution trend. As the time evolves, the remaining caching space with $\{60\text{MB}, 70\text{MB}\}$ will vanish due to the improvement of space utilization. Correspondingly, the remaining caching space with 30MB will present an upward trend as time goes by.

Finally, Fig. 5 depicts the evolution of the caching policy at the mean-field equilibrium, where each EDP can determine its optimal caching strategy $x_{i,k}$. When we choose a certain time slot, we observe that the optimal caching strategy will increase along with the growth of the caching state. This phenomenon is consistent with real-world scenarios, since each EDP will cache more contents when the remaining caching space is sufficient. Afterwards, we choose the caching states from $[10, 50]$ with a step of 10 and then observe the trend of the caching strategy under different caching states. As the time evolves, the EDP gradually decreases its own caching rate when the caching space is small (e.g., $q_k(t) = 10$), which can demonstrate the caching efficacy of MFG-CP. Certainly, the optimal caching strategy will continue to increase as long as there is sufficient remaining caching space.

2) *Impact of Parameters:* As depicted in Fig. 6, we provide the heap map description of the mean-field distribution under different Q_k values. It is notable that the caching space will gradually reach saturation with the increase of Q_k. This phenomenon occurs because the corresponding caching strategy will grow according to Eq. (21). To demonstrate the robustness of our algorithm, we further alter the standard deviation of the normal distribution in Fig. 7, which indicates the dispersion of the EDP's initial caching state. When we decrease the values of variance from 0.1^2 to 0.05^2, the heat map displays more concentrated results, i.e., the caching states among EDPs will be closer. By modifying the different values of Q_k, the observed trend in Fig. 7 is similar to that in Fig. 6.

Then, we vary the control coefficient w_5 of Eq. (8) in the range of $[0.65, 1.55] * 10^8$, as depicted in Fig. 8. Here, we abbreviate the value of w_5 in the legend for clarity. We

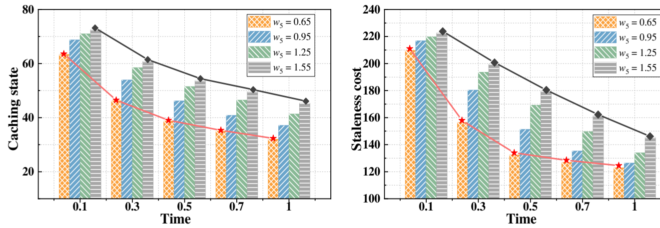


Fig. 8. Caching state and staleness cost vs. the control coefficient w_5.

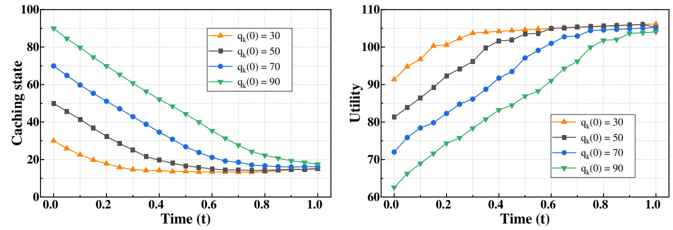


Fig. 9. Caching state and utility vs. the initial caching state.

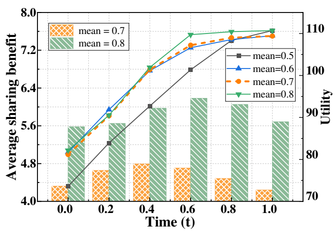


Fig. 10. Impact of initial distribution.

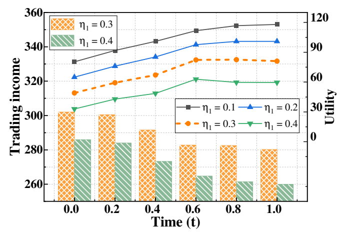


Fig. 11. Impact of the parameter η_1.

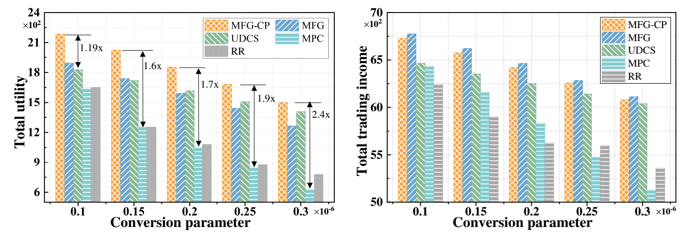


Fig. 12. Comparisons under different conversion parameters.

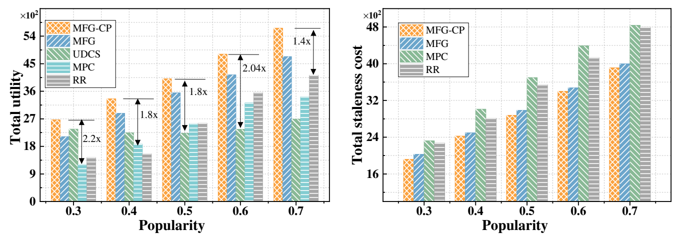


Fig. 13. Comparisons under different content popularity.

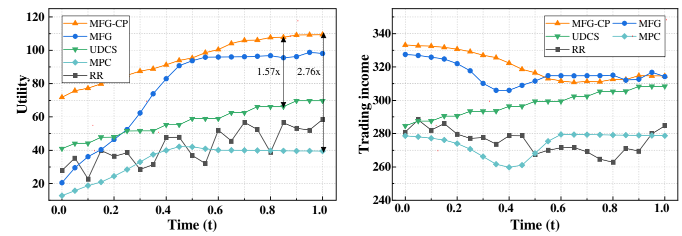


Fig. 14. Comparisons on utility and trading income.

observe that a smaller w_5 will reduce the caching state more slowly. The reason is that the EDP would opt to lower its caching strategy when the content placement cost is large. Meanwhile, it is also clear that a larger w_5 will lead to a higher staleness cost, since the EDP needs to spend more time acquiring contents from the center or other EDPs.

Next, we verify the convergence of MFG-CP by observing the caching state and the utility of an EDP. We randomly select a content k and set different initial caching states $q_k(0)$ in the range of $[30, 90]$. From Fig. 9, we see that the utility of $q_k(0)$ is lowest at first. This is because a larger initial caching state means that the EDP needs to spend more time caching more contents. In addition, we find that the remaining caching space and the utility of an EDP gradually tend towards stability, i.e., the EDP reaches an equilibrium state.

Finally, we evaluate the impact of the initial distribution and the conversion parameter η_1. As presented in Fig. 10, we initialize $\lambda(0)$ following a normal distribution with mean values from $\{0.5, 0.6, 0.7, 0.8\}$, and then assess both the utility of an EDP and the average sharing benefit brought by the mean-field group. We find that the average sharing benefit will have a slight fluctuation while the utilities can achieve stability. After adjusting the values of η_1 in the range of $\{0.1, 0.2, 0.3, 0.4\} * 10^{-6}$, we observe that the utility will gradually increase while the trading income has a gradual decrease over time. The reason is that many EDPs have cache enough contents and the trading processes between EDPs will be reduced. Moreover, a larger η_1 corresponds to a smaller utility and a lower trading income. This is because the trading price increases in accordance with Eq. (5).

TABLE II
COMPARISONS ON COMPUTATION TIME (SECOND)

Methods \ Number	50	100	200	300
MFG-CP	0.4319	0.4442	0.4336	0.5121
RR	0.1697	0.5527	0.9766	1.7832
MPC	0.1657	0.3157	0.8694	1.7094

3) *Performance Comparisons:* We first analyze the total utility and total trading income of an EDP under various values of η_1, as depicted in Fig. 12. Here, the accumulative utility/trading income is calculated over the finite time horizon. It is evident that improving the value of the conversion parameter leads to a reduction in total utility, which is consistent with the trend illustrated in Fig. 11. Notably, the total utility of MFG-CP surpasses that of MFG, UDCS, MPC, and RR, which can directly corroborate the superiority of our proposed framework. On the other hand, we find that the total trading income of MFG-CP is lower than that of MFG. This phenomenon is reasonable as EDPs can sell entire contents downloaded from the center when they cannot share contents with each other. Nonetheless, the staleness cost of MFG obviously exceeds that of MFG-CP. Hence, the behavior of content sharing is beneficial for improving the utility of each EDP, and our proposed MFG-CP framework enables each EDP to possess a higher utility than these compared algorithms.

Subsequently, we compare the five schemes by varying the popularity of content k within the range of $[0.3, 0.7]$. Here, we assume that the popularity is fixed during a certain time. Fig. 13 shows that MFG-CP exhibits a higher utility and a lower staleness cost compared to other baselines. This is because MFG-CP can make the best response even when dealing with system dynamics and information unpredictability. Particularly, UDCS holds the minimal variations in utility values under different content popularity and ignores the staleness cost. The reason is that UDCS focuses on minimizing the long-run average cost when considering overlapping contents and aggregate interference. Additionally, a higher Π_k brings in a higher utility owing to the growth of requests. In essence, caching more popular contents for EDPs will contribute to gaining more profits, which is in line with our design concept.

Next, we compare the utility and trading income of an EDP under different schemes, as depicted in Fig. 14. We observe that the utility of MFG-CP surpasses that of the compared algorithms. Notably, the utility of MFG-CP is 2.76 times and 1.57 times higher than that of MPC and UDCS, respectively. The reason is that RR and MPC do not consider the mutual influence among EDPs, and UDCS can reduce its cost while ignoring the economic competition among EDPs. Similar to Fig. 12, although there is a small gap in the trading income between MFG-CP and MFG, the staleness cost of MFG-CP is lower. Consequently, our proposed framework offers significant advantages in maximizing the utilities of EDPs.

Lastly, Table II presents the computation time of the three methods under different numbers of EDPs. Here, we omit the comparison of MFG and UDCS since their runtime is close to MFG-CP. We find that the advantages of MFG-CP become gradually pronounced as the number of EDPs increases. This is because the RR scheme requires M iterations of random number generation operations, whereas MFG-CP directly analyzes the average characteristics of the entire population rather than individual EDPs. In other words, the computational complexity of MFG-CP does not increase with the number of EDPs. This observation further validates the efficiency and scalability of MFG-CP in large-scale MEC systems.

VI. Related Works

We review the related works from the following aspects:

Mobile Edge Caching and Pricing: Several studies have paved the way for efficient edge caching solutions [15]–[18], [39]–[44]. For example, Sun *et al.* [16] developed an edge cache deployment strategy based on the prediction of users' preferences. Zong *et al.* [40] studied cocktail edge caching via deep reinforcement learning. Liu *et al.* [41] devised a dynamic caching replacement mechanism with privacy constraints and unobservable popularity. Meanwhile, great effort has been devoted to the pricing issue in MEC [45]–[47], e.g., Xiao *et al.* [45] proposed a reputation management scheme to maximize the revenue of vehicles in data trading. Huang *et al.* [46] designed a game-based profit sharing mechanism to decide the optimal revenue sharing ratio. However, these studies primarily focus on single-role scenarios, where caching and pricing are treated separately for all contents. Especially, when introducing content sharing into MEC systems, determining caching strategies and trading prices becomes more challenging. Only a handful of studies take caching and pricing into account [20]–[25], [33]. For instance, Li *et al.* [21] tackled the cache placement problem with auction-based trading to maximize the social welfare. Zou *et al.* [22] found the video pricing and the cache placement strategy by exploiting the Stackelberg game. Yang *et al.* [25] proposed a blockchain incentive scheme to determine the price vector and caching strategies, involving only the interaction between an EDP and users. Nevertheless, these works establish the homogeneity price with complete information and ignore the strategic behaviors of EDPs, so they are unsuitable for large-scale dynamic MEC systems.

Mean-Field Game: The MFG theory has drawn widespread attention across various large-scale systems, such as ultra-dense networks [26], [48], multi-access networks [49], [50], federated learning [51], the medical field [52], and so on. For instance, Narasimha *et al.* [26] utilized the MFG approach to identify the optimal probing strategies for devices in ultra-dense wireless networks. Shi *et al.* [50] proposed an MFG-guided task placement method by employing deep reinforcement learning so as to effectively reduce decision-making delays in task placement. However, these studies primarily focus on determining control strategies and do not consider the dynamics of EDPs and requesters. Only a few works study the instability of requesters and caching state [27], [28]. For example, Feng *et al.* [27] addressed a caching control problem with time-varying content requests to achieve cost minimization. Kim *et al.* [28] developed a spatio-temporal popularity dynamics model to minimize the long-run average cost. Nevertheless, these works aim at optimizing caching performance from the system perspective, without capturing the strategic and economic interplay among EDPs. In a nutshell, none of the existing works take strategic caching, economic content sharing, and dynamic pricing into account together, which involves a complex incomplete information game without knowing other players' states.

VII. Conclusion

In this paper, we study the competitive content placement issue in large-scale dynamic MEC systems. We first model the problem as a non-cooperative stochastic differential game, which considers the heterogeneous content demands, real-time trading prices, and paid content sharing. To facilitate decentralized decision-making, we propose the MFG-CP framework for joint content caching and pricing. Meanwhile, we develop an iterative best response learning scheme to determine the optimal caching strategy for each EDP. We also prove that these optimal strategies form a unique NE. Finally, extensive experiments validate the great performance of MFG-CP.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62172386, 61872330, and 61572457, the Natural Science Foundation of Jiangsu Province in China under Grant Nos. BK20231212, BK20211307, and BK20191194, and the National Natural Science Foundation of USA under Grant Nos. CNS 2128378, CNS 2107014, CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, and CNS 1651947. Mingjun Xiao is the corresponding author.

References

[1] J. Yao, S. Zhang, Y. Yao, F. Wang, J. Ma, J. Zhang, Y. Chu, L. Ji, K. Jia, T. Shen, A. Wu, F. Zhang, Z. Tan, K. Kuang, C. Wu, F. Wu, J. Zhou, and H. Yang, "Edge-cloud polarization and collaboration: A comprehensive survey for AI," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6866–6886, 2023.

[2] B. Glasbergen, K. Langendoen, M. Abebe, and K. Daudjee, "Chronocache: Predictive and adaptive mid-tier query result caching," in *ACM SIGMOD*, 2020, pp. 2391–2406.

[3] B. Li, Q. He, L. Yuan, F. Chen, L. Lyu, and Y. Yang, "Edgewatch: Collaborative investigation of data integrity at the edge based on blockchain," in *ACM SIGKDD*, 2022, pp. 3208–3218.

[4] W. Wingerath, F. Gessert, E. Witt, H. Kuhlmann, F. Bücklers, B. Wollmer, and N. Ritter, "Speed kit: A polyglot & gdpr-compliant approach for caching personalized content," in *IEEE ICDE*, 2020, pp. 1603–1608.

[5] J. Xie, D. Guo, X. Shi, H. Cai, C. Qian, and H. Chen, "A fast hybrid data sharing framework for hierarchical mobile edge computing," in *IEEE INFOCOM*, 2020, pp. 2609–2618.

[6] H. Zhou, T. Wu, H. Zhang, and J. Wu, "Incentive-driven deep reinforcement learning for content caching and D2D offloading," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2445–2460, 2021.

[7] X. Li, X. Wang, P. Wan, Z. Han, and V. C. M. Leung, "Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1768–1785, 2018.

[8] M. Dehghan, B. Jiang, A. Seetharam, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. K. Sitaraman, "On the complexity of optimal request routing and content caching in heterogeneous cache networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1635–1648, 2017.

[9] L. Wang, S. Chen, and Q. He, "Concept drift-based runtime reliability anomaly detection for edge services adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 153–12 166, 2023.

[10] C. Chen, Y. Yuan, Z. Wen, G. Wang, and A. Li, "GQP: A framework for scalable and effective graph query-based pricing," in *IEEE ICDE*, 2022, pp. 1573–1585.

[11] Google edge network. [Online]. Available: https://peering.google.com

[12] Fastly–edge cloud platform. [Online]. Available: https://www.fastly.com/edge-cloud-platform

[13] CacheMire–wireless edge caching platform. [Online]. Available: https://cordis.europa.eu/project/rcn/209160/factsheet/en

[14] Vbrick edge caching. [Online]. Available: https://vbrick.com/edge-caching

[15] R. Aghazadeh, A. Shahidinejad, and M. Ghobaei-Arani, "Proactive content caching in edge computing environment: A review," *Software: Practice and Experience*, vol. 53, no. 3, pp. 811–855, 2023.

[16] H. Sun, Y. Chen, K. Sha, S. Huang, X. Wang, and W. Shi, "A proactive on-demand content placement strategy in edge intelligent gateways," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 7, pp. 2072–2090, 2023.

[17] D. Yu, T. Wu, C. Liu, and D. Wang, "Joint content caching and recommendation in opportunistic mobile networks through deep reinforcement learning and broad learning," *IEEE Transactions on Services Computing*, pp. 1–14, 2023.

[18] O. M. Dung, M. Chen, T. Taleb, X. Wang, and V. C. M. Leung, "FGPC: fine-grained popularity-based caching design for content centric networking," in *ACM MSWiM*, 2014, pp. 295–302.

[19] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1227–1240, 2020.

[20] Y. Song, L. Jiao, R. Yang, T. Wo, and J. Xu, "Incentivizing online edge caching via auction-based subsidization," in *IEEE SECON*, 2022, pp. 253–261.

[21] C. Li, S. Liang, J. Zhang, Q. Wang, and Y. Luo, "Blockchain-based data trading in edge-cloud computing environment," *Information Processing & Management*, vol. 59, no. 1, p. 102786, 2022.

[22] J. Zou, C. Li, C. Zhai, H. Xiong, and E. G. Steinbach, "Joint pricing and cache placement for video caching: A game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1566–1583, 2019.

[23] F. Tütüncüoğlu and G. Dán, "Optimal service caching and pricing in edge computing: a bayesian gaussian process bandit approach," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2022.

[24] B. Jedari and M. D. Francesco, "Auction-based cache trading for scalable videos in multi-provider heterogeneous networks," in *IEEE INFOCOM*, 2019, pp. 1864–1872.

[25] Y. Yang, Z. Liu, Z. Liu, Y. Xie, K. Y. Chan, and X. Guan, "Joint optimization of edge computing resource pricing and wireless caching for blockchain-driven networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6661–6670, 2022.

[26] D. Narasimha, S. Shakkottai, and L. Ying, "Age-dependent distributed MAC for ultra-dense wireless networks," in *IEEE INFOCOM*, 2021, pp. 1–10.

[27] H. Feng, S. Guo, D. Liu, and Y. Yang, "Mean-field game theory based optimal caching control in mobile edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2022.

[28] H. Kim, J. Park, M. Bennis, S. Kim, and M. Debbah, "Mean-field game theoretic edge caching in ultra-dense networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 935–947, 2020.

[29] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese journal of mathematics*, vol. 2, no. 1, pp. 229–260, 2007.

[30] Y. Jiang, Y. Hu, M. Bennis, F. Zheng, and X. You, "A mean field game-based distributed edge caching in fog radio access networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1567–1580, 2020.

[31] Q. Wang, S. Guo, J. Liu, C. Pan, and L. Yang, "Motishare: Incentive mechanisms for content providers in heterogeneous time-varying edge content market," *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 452–465, 2023.

[32] Y. Xu, M. Xiao, Y. Zhu, J. Wu, S. Zhang, and J. Zhou, "Aoi-guaranteed incentive mechanism for mobile crowdsensing with freshness concerns," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2023.

[33] C. Jiang, L. Gao, J. Luo, and et al., "A game-theoretic analysis of joint mobile edge caching and peer content sharing," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1445–1461, 2023.

[34] J. Huang and L. Gao, *Wireless Network Pricing*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2013.

[35] Y. Xu, M. Xiao, J. Wu, H. Tan, and G. Gao, "A personalized privacy preserving mechanism for crowdsourced federated learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1568–1585, 2024.

[36] L. C. Evans, *Partial differential equations*. American Mathematical Society, 2022, vol. 19.

[37] P. E. Caines, M. Huang, and R. P. Malhamé, "Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle," *Communications in Information and Systems*, vol. 6, no. 3, pp. 221–252, 2006.

[38] YouTube. Trending YouTube Video Statistics. (2023, May 20). [Online]. Available: https://www.kaggle.com/datasets/datasnaek/youtube-new

[39] D. Dash, V. Kantere, and A. Ailamaki, "An economic model for self-tuned cloud caching," in *IEEE ICDE*, 2009, pp. 1687–1693.

[40] T. Zong, C. Li, Y. Lei, G. Li, H. Cao, and Y. Liu, "Cocktail edge caching: Ride dynamic trends of content popularity with ensemble learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 208–219, 2023.

[41] S. Liu, C. Zheng, Y. Huang, and T. Q. S. Quek, "Distributed reinforcement learning for privacy-preserving dynamic edge caching," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 749–760, 2022.

[42] T. Liu, S. Ni, X. Li, Y. Zhu, L. Kong, and Y. Yang, "Deep reinforcement learning based approach for online service placement and computation resource allocation in edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3870–3881, 2023.

[43] S. An, Y. Cao, and W. Zhao, "Competitive consistent caching for transactions," in *IEEE ICDE*, 2022, pp. 2154–2167.

[44] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, and M. Wang, "Low-rank multi-view embedding learning for micro-video popularity prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1519–1532, 2018.

[45] H. Xiao, L. Cai, J. Feng, Q. Pei, and W. Shi, "Resource optimization of mab-based reputation management for data trading in vehicular edge computing," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5278–5290, 2023.

[46] Y. Huang, Y. Zeng, F. Ye, and Y. Yang, "Profit sharing for data producer and intermediate parties in data trading over pervasive edge computing

environments," *IEEE Transactions on Mobile Computing*, vol. 22, no. 1, pp. 429–442, 2023.

[47] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *IEEE ICDE*, 2016, pp. 49–60.

[48] D. Wang, W. Wang, Y. Kang, and Z. Han, "Distributed data offloading in ultra-dense LEO satellite networks: A stackelberg mean-field game approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 112–127, 2023.

[49] Y. Kang, H. Wang, B. Kim, J. L. Xie, X. Zhang, and Z. Han, "Time efficient offloading optimization in automotive multi-access edge computing networks using mean-field games," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 6460–6473, 2023.

[50] D. Shi, H. Gao, L. Wang, M. Pan, Z. Han, and H. V. Poor, "Mean field game guided deep reinforcement learning for task placement in cooperative multiaccess edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9330–9340, 2020.

[51] Y. Kang, Y. Zhu, D. Wang, Z. Han, and T. Başar, "Joint server selection and handover design for satellite-based federated learning using mean-field evolutionary approach," *IEEE Transactions on Network Science and Engineering*, 2023.

[52] J. Huang, H. Yan, J. Li, H. M. Stewart, and F. C. Setzer, "Combining anatomical constraints and deep learning for 3-d CBCT dental image multi-label segmentation," in *IEEE ICDE*, 2021, pp. 2750–2755.