Joint Pricing and Cache Placement for Video Caching: A Game Theoretic Approach

Junni Zou, *Member, IEEE*, Chenglin Li, *Member, IEEE*, Congcong Zhai, Hongkai Xiong, *Senior Member, IEEE*, and Eckehard Steinbach, *Fellow, IEEE*

***Abstract*— Caching can effectively smooth the temporal traffic variability and decrease the redundant data transmission in mobile video delivery. In this paper, we consider a video caching system consisting of a video provider (VP), a mobile network operator (MNO) with a set of cache-enabled base stations (BSs), and multiple mobile users. The VP leases some popular videos to the MNO, while the MNO places these rented videos in local caches of its BSs to save expensive backhaul transmission cost. However, in such a two-sided market, these two entities are competing with each other for their own profit due to their opposite expectation on the video pricing. To address this, we model the competition between the two entities using the framework of Stackelberg games and propose a joint video pricing and cache placement strategy by considering the heterogeneity of video file sizes and exploiting the classic law of demand from the field of economics. The proposed optimization problem is able to jointly maximize the profit of the VP and the MNO by the optimal selection of the video pricing and the cache placement strategy given that price, for both noncooperative BS caching and cooperative BS caching cases. We then develop iterative algorithms based on dynamic programming and gradient ascent, respectively, for these two cases to find the Stackelberg equilibrium (SE). The simulation results further show that the proposed joint optimization formulation follows the law of demand in economics, and the proposed algorithms for both cases can efficiently converge to the SE point that jointly maximizes the profit for both the VP and the MNO.**

***Index Terms*— Wireless caching, video pricing, mobile video delivery, cache placement, Stackelberg game.**

I. INTRODUCTION

WITH the ever-increasing and widespread use of smart devices, wireless video streaming is currently experiencing extensive growth and being leveraged for a wide range of multimedia applications, such as mobile multimedia services. Statistics show that mobile video data has become the dominant source for the exponential growth of global mobile data traffic over cellular networks [1]. A noticeable consequence of the extensive growth of mobile video traffic is the resultant acceleration of busy-hour traffic compared to the average traffic growth. Unlike general web usage occurring throughout the whole day, video usage is more prone to be requested and consumed during the evening hours. Consequently, more video usage means more traffic during the peak hours of the day. Therefore, the mobile video traffic presents a usually high temporal variability, which causes traffic congestion during peak hours yet network under-utilization during off-peak hours. On the other hand, most of the streaming requests of mobile users are repetitive and concentrated on some popular videos, which incurs redundant data transmission over the network [2].

The temporal variability and redundancy of video transmissions can be efficiently reduced by locally caching some of the popular videos in the storage of intermediate network nodes [3]. In general, the wireless video caching problem comprises two separate phases [4]. The first phase is the placement phase, which includes pre-fetching videos in the local cache and is performed usually during off-peak hours when the network resource is abundant or before the start of video streaming services. The second phase is the delivery phase in which requested videos are delivered to mobile users either from the local cache or through the expensive backhaul connections. Studies to date have investigated work related to the aforementioned cache placement and delivery from different perspectives. A fundamental information-theoretic formulation of the general caching problem is proposed in [4], which is able to reduce the overall data transmission in the delivery phase through the joint optimization of the placement and delivery phases to achieve an important caching gain. For mobile video traffic, caching at distributed local caches that are placed geographically closer to the mobile users has been demonstrated capable of significantly relieving the video traffic load of the mobile operator's network, by replacing the usually weak yet expensive backhaul links with the high-speed local connections to ensure the low delay requirement of users [3]. To further address the heterogeneity issue of the network and mobile users, a mobile edge cache placement framework [5] is proposed for dynamic adaptive video streaming, in order to maximize the users' quality-of-experience (QoE) by considering the coordination among local caches and different rate-distortion behavior of videos.

Manuscript received October 12, 2018; revised March 15, 2019; accepted April 21, 2019. Date of publication May 13, 2019; date of current version June 17, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61622112, Grant 61871267, Grant 61831018, Grant 61529101, Grant 61425011, and Grant 61720106001 and in part by the Alexander von Humboldt Foundation. *(Corresponding author: Chenglin Li.)*

J. Zou, C. Li, and H. Xiong are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zou-jn@cs.sjtu.edu.cn; lcl1985@sjtu.edu.cn; xionghongkai@sjtu.edu.cn).

C. Zhai was with the Department of Communication Engineering, Shanghai University, Shanghai 200072, China. He is now with Meituan-Dianping, Shanghai, China (e-mail: yujiaerzong@t.shu.edu.cn).

E. Steinbach is with the Chair of Media Technology, Technical University of Munich, 80333 Munich, Germany (e-mail: eckehard.steinbach@tum.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2019.2916279

0733-8716 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

All of these caching schemes mainly focus on the optimal cache placement from the network and user perspective, with the purpose of either reducing the traffic load of the network or enhancing the users' QoE. However, mobile multimedia services today are tightly coupled with economic issues, such as pricing for videos at the content provider and rental of storage at the network service provider. To investigate the caching problem from such an economic perspective, the joint pricing and caching problem in [6] considers a small-cell video caching system where multiple mobile network operators (MNOs) rent some popular videos from a video provider (VP) and store these videos in the local cache of their small-cell base stations (BSs), and jointly maximizes the profit of the MNOs and the VP through a Stackelberg game. In [7], a joint pricing and resource allocation framework is formulated for a similar system, where multiple VPs rent the storage space from an MNO for caching their videos. However, the limitation of these schemes is that the optimal decision variable for caching in their framework is the fraction of BSs that cache a specific video, while the specific cache placement strategy stating the caching decision between BSs and videos is still unclear. In addition, their joint pricing and caching decision is based on a simplified assumption of homogeneous video file size. This assumption may become infeasible in practical video streaming applications, since the output video files after the encoding/compression may have heterogeneous sizes due to different rate-distortion behavior.

To address the above issues, in this paper, we propose a joint video pricing and cache placement optimization framework for the video caching system with proper consideration of the heterogeneity of video file sizes and by exploiting the classic law of demand from the field of economics. Specifically, we consider a video caching system consisting of a video provider (VP), a mobile network operator (MNO) with a set of cache-enabled base stations (BSs) and multiple mobile users. This video caching system is further viewed as a two-sided market composed of the VP and the MNO. In such a two-sided market, the MNO saves the backhaul transmission cost by renting some popular videos from the VP and caching the copies of these videos in the local storage of its BSs, while the VP also makes some profit via leasing these videos. However, the law of demand in economics reveals that the objectives of maximizing their own profit for these two entities are conflicting since they have opposite expectation on the video pricing. We therefore model the competition between the VP and the MNO as a single-leader single-follower Stackelberg game to jointly maximize the profit of the VP and the MNO, for both the noncooperative BS caching case and the cooperative caching through coordination between BSs.

For the noncooperative caching case where each BS independently makes the caching decision, we show that the MNO's profit maximization problem can be decoupled as a set of knapsack problems with respect to the actual storage space of each BS. We then develop an iterative dynamic programming algorithm to efficiently find the Stackelberg Equilibrium (SE) point of the proposed Stackelberg game, which corresponds to the optimal video pricing for the VP and the optimal cache placement strategy for the BSs of the MNO. For the case where BSs are cooperating, we employ the split cache strategy optimally designed for cooperative caching in the proposed Stackelberg game. We then prove that the MNO's profit maximization problem is reformulated as a convex optimization problem, and develop an iterative gradient ascent algorithm to find the SE point. We conduct extensive simulations under different system settings. Simulation results demonstrate that the proposed optimization formulation follows the law of demand in economics, and the proposed algorithms for both cases can efficiently converge to the SE point that jointly maximizes the profit of the VP and the MNO. Simulation results also show that their profit can be further increased by allowing the coordination between BSs.

The rest of this paper is organized as follows. Section II reviews the related works in the literature. In Section III, we introduce the joint video pricing and cache placement framework for the video caching system and related system models. In Section IV, we model the profit gained by the MNO and the VP in the video caching system, and generally formulate a joint pricing and cache placement optimization problem for both noncooperative and cooperative caching under the framework of Stackelberg game. In Sections V and VI, we develop an iterative dynamic programming algorithm and an iterative gradient ascent algorithm to find the SE point, respectively, for both cases. Section VII presents simulation results, and evaluates the gain of the proposed algorithms compared to existing algorithms. We conclude this paper in Section VIII.

II. Related Work

Many works have been done to leverage caching in different multimedia data delivery systems by designing various caching strategies to improve the cache utilization [3], [5], [8]–[12]. A joint routing and caching problem is formulated in [8] to obtain the optimal fraction of content requests locally served by distributed caches, subject to constraints such as the bandwidth of the network, the storage capacity of local caches, and the content request patterns of users. Chen *et al.* [9] utilize caching in the wireless device-to-device network, and propose an incentive mechanism to encourage the mobile users to share their contents with the optimal rewarding policy obtained through a Stackelberg game approach. For cooperative content caching, Shanmugam *et al.* [3] propose an optimal caching placement scheme for cooperative local caches to minimize the average downloading delay experienced by mobile users. The split cache strategy that divides the cache storage space into a duplicate partition and a unique partition is proposed for the distributed cooperative caching in [10], to provide the optimal object placement with minimal overall download cost in the social wireless network. This strategy is theoretically proved and numerically analyzed as the optimal cache placement strategy for networks with homogeneous assumptions about content demands, cache capacities and content sizes. Jiang *et al.* [11] formulate a joint cache placement and content delivery optimization problem for cooperative caching over heterogeneous cellular networks, and develop a hierarchical primal-dual approach to find the optimal caching and delivery policy. To utilize caching in adaptive video streaming systems, an adaptive caching strategy for information centric networks is proposed in [12], which adapts to variable video contents with heterogeneous bitrates and is able to reduce the access delay for the requested bitrate of each user. The adaptive video streaming is also leveraged in the mobile edge cache placement framework [5], which maximizes the users' QoE by considering the coordination among local caches and different rate-distortion behavior of videos. To obtain the optimal cache placement strategy, a polynomial-time greedy algorithm is developed with theoretical proof on the performance approximation bound. As mentioned, all of these works focus on the optimal cache placement from the network's or the user's perspective, and neglect the economic issues that might affect the caching strategy design in the two-sided market between the content provider and the network.

In another line of research, the idea of investigating the caching strategy from an economic perspective has been recently explored in [6], [7], [13]–[16]. Through a careful evaluation of the trade-off between the cache storage, edge server transcoding, and backhaul transmission cost, Jin *et al.* [13] integrate caching with transcoding operations in the media cloud, and analyze the optimal caching and transcoding resource allocation scheme to minimize the total operational cost of the system for the noncooperative caching case. Similarly, the trade-off between cache storage cost and transcoding computation cost is investigated for the media cloud [14], and for the video-on-demand systems with multiple video versions [15], to develop a cost-efficient caching and transcoding strategy. Li *et al.* [6] consider a small-cell video caching system where multiple MNOs rent popular videos from a video provider (VP) and cache these videos in their small-cell BSs, and propose a joint pricing and caching scheme to maximize the profit of the MNOs and the VP through a Stackelberg game. The authors in [16] further extend the joint pricing and caching policy design to the information centric network. Similarly in [7], a joint pricing and resource allocation framework is formulated, where multiple VPs rent the storage space from an MNO for caching their videos.

This work differs from the related literature in the following aspects. First, we study the optimal video pricing and cache placement strategy with proper consideration of the heterogeneity of video file sizes, which is a more practical scenario for compressed videos but is not considered in most of the existing literature. Second, we employ the law of demand from the field of economics to explain the proposed Stackelberg game and to strike the balance between the video pricing decision made by the VP and the actual demand for caching from the MNO, and also demonstrate this law through our simulation results. Third, we reveal explicitly the optimal cache placement strategy to state the caching decision between BSs and videos for both noncooperative and cooperative caching, which is not shown in most of the joint pricing and caching schemes.

III. Framework and System Models

In this section, we introduce the joint video pricing and cache placement framework for the video caching system and related system models.

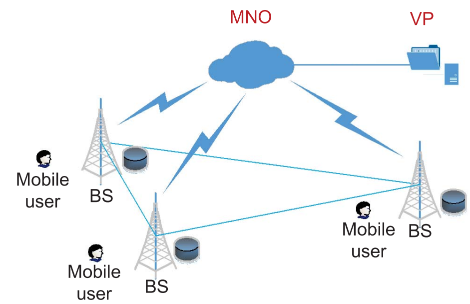

Fig. 1. A video caching system comprising a VP and an MNO with cache-enabled BSs and mobile users.

A. Framework

As illustrated in Fig. 1, we consider a video caching system comprising a video provider (VP), a mobile network operator (MNO) with a set of cache-enabled base stations (BSs), and multiple mobile users. To reduce the mobile video traffic over the backhaul links between the VP and BSs, and to enable faster video delivery to the users, the MNO is willing to rent some popular videos from the VP and cache the copies of these videos in the local storage of its BSs that are geographically closer to the users. Through leasing these video copies to the MNO, on the other hand, the VP also makes some profit. Therefore, both the VP and the MNO can benefit from the local caching system. However, they are both assumed to be selfish and intend to maximize their own revenues.

The overall framework of the proposed video caching system is as follows. The VP publishes some popular videos on its website after purchasing the copyrights of these videos. The MNO then bargains with the VP about the unit rental cost that the VP charges for renting a video. The objective of the joint video pricing and cache placement is to reach the optimal pricing scheme for the VP and the optimal caching strategy for each BS of the MNO under that price, which corresponds to the equilibrium between the two competing entities within the game theoretic framework. Upon reaching the equilibrium, the MNO will place the rented video copies into the local storage of its BSs following the optimal caching scheme. The mobile users will connect to their adjacent BSs for downloading the desired videos. If a copy of the requested video is found in the local cache of the BS directly connecting to the user (or any other indirectly connected BS when the BSs are cooperating with each other), the MNO will send the cached copy to the user with a low downloading delay and transmission cost. Otherwise, the BS has to first request and obtain the video from the VP via the possibly congested and weak backhaul links, and then send the video to the user, which results in a much higher downloading delay and transmission cost.

B. Network Model

We now describe in more detail the models considered in this work, and introduce the notations.

On the VP side, let first $\mathcal{F} = \{f_1, f_2, \ldots, f_J\}$ denote the set of J video files that are published by the VP. We further assume that this set is sorted in decreasing order of popularity, i.e., f_j represents the j-th popular video out of the set $\mathcal{F}$. To improve the transmission efficiency, these videos are compressed and encoded by a standardized video encoder (e.g., H.264, HEVC) before transmission. In practice, due to the difference of video types (e.g., movies, sport videos, cartoon, etc.) and video contents (e.g., motion of the objects) among these videos, the output video files after compression may have heterogeneous sizes. In this regard, s_j is used to denote the file size of video $f_j \in \mathcal{F}$.

For the MNO, let $\mathcal{B} = \{b_1, b_2, \ldots, b_I\}$ denote the set of I BSs deployed between the VP and mobile users, where each BS b_i has a storage capacity of S_i. We denote by c^{bh} the unit transmission cost of the backhaul links between the VP and the BSs, by c^{co} the unit transmission cost of the links between any two BSs, and by c^{mu} the unit transmission cost of the links connecting the mobile users to the BSs. The unit transmission delay of a link (m, n) is denoted as $d_{m,n}$.

C. Video Request Generation Model

In accordance with [3], [13], [17], we study here the video caching system with the optimal decision of pricing and cache placement to be made for a certain period of time. We therefore assume the average demand during this time period for the set of J video files to be known in advance, and adopt the same assumption from [3], [13] that the video requests from mobile users are statistically independent. Such an independent user request model is an acceptable approximation in the average sense or for the case when the content popularity variation over time is relatively slow.

To model the independent user request, we use a probability mass function P_j to denote the average probability of a video file $f_j \in \mathcal{F}$ being requested by mobile users. Specifically, as a widely used distribution for the popularity based request modeling [18], the Zipf distribution is adopted to model the popularity distribution of video files (i.e., the distribution of video request possibilities). With this model, the average probability of requesting the j-th popular video file f_j is inversely proportional to its rank and given by:

$$P_j = \frac{\Omega}{j^{\alpha}}, \Omega = \frac{1}{\sum_{j=1}^{J} 1/j^{\alpha}}, \quad \forall f_j \in \mathcal{F}, \tag{1}$$

where the Zipf parameter $\alpha \in [0, 1]$ determines the skewness of the popularity distribution. A larger value of α indicates a more concentrated request distribution, i.e., the majority of requests are made by the users for the first few popular video files. It can also be seen from Eq. (1) that the video file set is sorted in a decreasing order of popularity, where a smaller index j corresponds to a higher popularity P_j.

D. Pricing Model and the Law of Demand

In the field of economics [19], [20], the relationship between the price and demand of a good is concluded by the law of demand. It claims that when other factors are kept equal, the demanded quantity of a good falls as the price of the good rises. Following this law, the demand versus price curve can be characterized as a downward-sloping line.

In the video caching system in Fig. 1, the videos are considered as the goods, while each BS of the MNO as a buyer. The demand of BS $b_i \in \mathcal{B}$ is then defined as the amount of videos (normalized to the unit file size of these videos) that the BS is willing to purchase for renting and caching in its local cache, which is denoted as Q_i. In this paper, we adopt the revenue sharing contract model in [21]–[23] that is widely used in video rental industry. Under this contract, the VP charges a wholesale rental price per unit of video δ, and the payment from users for downloading and watching the cached videos in BSs of the MNO is divided between the VP and the MNO, with a revenue sharing percentage θ retained for the MNO. Then, based on the law of demand, when the VP raises the unit price δ for renting a video, the MNO will accordingly decrease the cached amount of videos in each BS to save the rental cost.

E. Cache Placement and Service Modes

We further consider a caching system where a video file is either fully cached or not cached at all in the local storage of any BS. The local cache placement strategy of the BSs can then be represented by a bipartite graph $\mathcal{G}_{b_i,f_j} = (\mathcal{B}, \mathcal{F}, \mathcal{E}_{b_i,f_j})$ between vertices representing the BSs in $\mathcal{B}$ and vertices describing the video files in $\mathcal{F}$. An edge $(b_i, f_j) \in \mathcal{E}_{b_i,f_j}$ is drawn when video file f_j is rented by and placed in the local cache of BS b_i. To better understand the cache placement strategy indicated by the bipartite graph, we can further denote $\boldsymbol{X}_{I\times J}$ as an $I \times J$ adjacency matrix of $\mathcal{G}_{b_i,f_j}$, such that $\forall b_i \in \mathcal{B}$ and $\forall f_j \in \mathcal{F}$, $x_{i,j} = 1$ indicates that an edge $(b_i, f_j) \in \mathcal{E}_{b_i,f_j}$ exists and $x_{i,j} = 0$ denotes the absence of an edge between b_i and f_j, i.e.,

$$x_{i,j} = \begin{cases} 1, & \text{if the BS } b_i \text{ caches the video file } f_j; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In this paper, we study the optimal cache placement strategy under two different base station service modes: the noncooperative caching and the distributed cooperative caching.

1) Noncooperative Caching: In the noncooperative caching case, the BSs are not communicating with each other. Whenever a mobile user that connects to BS b_i submits a playback request for a video file f_j, the MNO will first search in the local cache of BS b_i. If a copy of that video file is already cached, the BS will directly serve the user's request by sending the cached content with a local transmission cost of $s_j \cdot c^{mu}$ and a local transmission delay of $s_j \cdot d_{i,mu}$. In this case, the VP will also share its received user payment for that video with the MNO. Otherwise, the BS has to first request and obtain the video from the VP via the congested and weak backhaul link, and then send the video to the user, which incurs additionally a backhaul transmission cost of $s_j \cdot c^{bh}$ and a backhaul transmission delay of $s_j \cdot d_{vp,i}$. Taking into account the cache placement matrix $\boldsymbol{X}_{I\times J}$, the average revenue, in terms of the backhaul transmission cost plus the backhaul transmission delay cost saved by local caching for serving a user connecting to BS b_i with the requested video file f_j, can then be expressed as:

$$\bar{R}_{i,j}^{save} = s_j \cdot x_{i,j} \cdot (c^{bh} + \varsigma \cdot d_{vp,i}), \quad \forall f_j \in \mathcal{F}, \ \forall b_i \in \mathcal{B}, \tag{3}$$

where ς is a weight constant relating the delay cost to the transmission delay. It can be seen from Eq. (3) that if the video file f_j is cached in the local cache of BS b_i, then $x_{i,j} = 1$ and $\bar{R}_{i,j}^{save} = s_j \cdot (c^{bh} + \varsigma \cdot d_{vp,i})$; otherwise, $x_{i,j} = 0$ and $\bar{R}_{i,j}^{save} = 0$. On the other hand, the shared user payment incurred by local caching for serving a user connecting to BS b_i with the requested video file f_j can then be expressed as:

$$\bar{R}_{i,j}^{share} = x_{i,j} \cdot \theta \cdot r_j \quad \forall f_j \in \mathcal{F},\ \forall b_i \in \mathcal{B}, \tag{4}$$

where r_j is the price for video file f_j that is paid by the user to the VP, and θ is the revenue sharing percentage that the MNO will retain from the user's payment. It can be seen from Eq. (4) that if the video file f_j is cached in the local cache of BS b_i, then $x_{i,j} = 1$ and $\bar{R}_{i,j}^{share} = \theta \cdot r_j$; otherwise, $x_{i,j} = 0$ and $\bar{R}_{i,j}^{share} = 0$.

2) Distributed Cooperative Caching: In the distributed cooperative caching case, the BSs are able to communicate with their adjacent BSs. Here, we denote by $\mathcal{N}(b_i)$ the set of adjacent BSs to b_i, and assume that $\mathcal{N}(b_i)$ is sorted in an increasing order of the transmission delay to b_i, such that $b_{(l)_i} \in \mathcal{N}(b_i)$ represents the BS with the l-th smallest transmission delay to the BS b_i. In particular, we define the notation $b_{(0)_i} = b_i$ to denote the BS b_i itself. Whenever a playback request for a video file f_j is submitted by a mobile user, the MNO will also try to first serve the user by the cached copy in the local cache of the BS b_i that connects to the user. When the requested video file is not locally cached, BS b_i will communicate with its adjacent BSs according to the order of $\mathcal{N}(b_i)$, searching for the cached copy of that video. If at least one adjacent BS in $\mathcal{N}(b_i)$ caches the requested video, this determination will continue until a copy of that video is found in the local cache of an adjacent BS $b_k \in \mathcal{N}(b_i)$. In this case, BS b_i will obtain the video from the BS b_k and send the video to the user, which incurs additionally a transmission cost of $s_j \cdot c^{co}$ and a transmission delay of $s_j \cdot d_{k,i}$. When no copy of the requested video is available at the local cache of any adjacent BS, the BS has to request and obtain the video from the VP via the congested and weak backhaul link, and then send the video to the user. However, this will result in a much more expensive transmission cost (i.e., $c^{bh} \gg c^{co}$) and a much larger transmission delay (i.e., $d_{vp,i} \gg d_{k,i}$), since the backhaul communication resource is usually very limited compared to the high-speed links provided by the adjacent BSs. In the cooperative case, the average revenue saved by local caching for serving a user connecting to BS b_i with the requested video file f_j is given by:

$$\begin{aligned}\bar{R}_{i,j}^{save} = s_j \cdot \Bigg\{ &(c^{bh} + \varsigma \cdot d_{vp,i}) \\ &- \sum_{n=1}^{|\mathcal{N}(b_i)|} \left[\prod_{m=0}^{n-1} (1 - x_{(m)_i,j})\right] x_{(n)_i,j} (c^{co} + \varsigma \cdot d_{(n)_i,i}) \\ &- \left[\prod_{m=0}^{|\mathcal{N}(b_i)|} (1 - x_{(m)_i,j})\right] (c^{bh} + \varsigma \cdot d_{vp,i}) \Bigg\}, \\ &\forall f_j \in \mathcal{F},\ \forall b_i \in \mathcal{B}. \end{aligned} \tag{5}$$

Eq. (5) is derived in accordance with the proposed determination process for distributed cooperative caching. In Eq. (5), if the video file f_j is cached in the local cache of BS b_i, then $x_{(0)_i,j} = 1$ and the last two terms equal to zero. In this case, we have $\bar{R}_{i,j}^{save} = s_j \cdot (c^{bh} + \varsigma \cdot d_{vp,i})$. Otherwise, either one of the last two terms in Eq. (5) becomes nonzero. If the BS $b_{(\hat{n})_i} \in \mathcal{N}(b_i)$ caches a copy of video file f_j and any other BS with a transmission delay lower than $b_{(\hat{n})_i}$ fails to cache this video, (i.e., $x_{(\hat{n})_i,j} = 1$ and $x_{(n)_i,j} = 0, \forall n < \hat{n}$), we then have the term $[\prod_{m=0}^{n-1}(1 - x_{(m)_i,j})]x_{(n)_i,j}$ equals to one for $n = \hat{n}$ and zero for any other adjacent BS $n \in \mathcal{N}(b_i) \setminus \{\hat{n}\}$, while the term $[\prod_{m=0}^{|\mathcal{N}(b_i)|}(1 - x_{(m)_i,j})] = 0$. If no copy of the video file f_j can be found in either the BS b_i or any of the adjacent BSs in $\mathcal{N}(b_i)$ (i.e., $x_{(0)_i,j} = 0$ and $x_{(n)_i,j} = 0, \forall n \in \mathcal{N}(b_i)$), the term $[\prod_{m=0}^{|\mathcal{N}(b_i)|}(1 - x_{(m)_i,j})] = 1$, while $\sum_{n=1}^{|\mathcal{N}(b_i)|}[\prod_{m=0}^{n-1}(1 - x_{(m)_i,j})]x_{(n)_i,j} = 0$.

Similarly, the shared user payment incurred by local caching for serving a user connecting to BS b_i with the requested video file f_j is given by:

$$\begin{aligned}\bar{R}_{i,j}^{share} = \ &\theta \cdot r_j - \left[\prod_{m=0}^{|\mathcal{N}(b_i)|} (1 - x_{(m)_i,j})\right] \cdot \theta \cdot r_j, \\ &\forall f_j \in \mathcal{F}, \quad \forall b_i \in \mathcal{B}. \end{aligned} \tag{6}$$

In Eq. (6), if no copy of the video file f_j can be found in either the BS b_i or any of the adjacent BSs in $\mathcal{N}(b_i)$, then the term $[\prod_{m=0}^{|\mathcal{N}(b_i)|}(1 - x_{(m)_i,j})] = 1$ and $\bar{R}_{i,j}^{share} = 0$; otherwise, the requested video f_j will be served directly by the BS with the cached copy, resulting in $\bar{R}_{i,j}^{share} = \theta \cdot r_j$.

IV. Stackelberg Game Formulation for Joint Pricing and Cache Placement Optimization

In this section, we model the profit gained by the MNO and the VP in the video caching system, and then generally formulate a joint pricing and cache placement optimization problem for both the noncooperative and cooperative caching cases under the framework of Stackelberg games.

A. Profit Modeling

The overall profit gained by the MNO through the local caching is determined by three components: the revenue of the saved cost (including both the transmission cost and the delay cost) for backhaul transmission, the revenue of the shared user payment gained by local caching in the BSs, and the video rental cost paid for renting some popular video files from the VP and caching them in these BSs.

As seen in Eqs. (3)-(6), for both the noncooperative and cooperative caching cases, the average revenue achieved by local caching for serving a user connecting to BS b_i with requested f_j can be derived and denoted as $\bar{R}_{i,j}^{save} + \bar{R}_{i,j}^{share}$. Therefore, the total revenue of the MNO gained by local caching in all its BSs is given by:

$$R_{MNO}(\boldsymbol{X}) = \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} P_j \lambda_i \cdot (\bar{R}_{i,j}^{save} + \bar{R}_{i,j}^{share}), \tag{7}$$

where λ_i is the expected number of video download requests submitted from the mobile users to BS b_i and $P_j \lambda_i$ thus denotes the expected number of requests submitted to b_i for video file f_j.

In order to cache video file f_j into the local storage of its BS b_i, the MNO has to rent a copy of that video from the VP with a rental cost of $s_j \cdot \delta$, where s_j represents the file size of f_j and δ is the unit wholesale rental price that the VP charges for renting a video. Here, the revenue sharing contract model in [21]–[23] is adopted with a further consideration on video sizes. The motivation is as follows. If we take a specific type of videos with similar R-D behavior and spatial resolutions for example and encode them with the same encoder parameters, the output video sizes are then approximately proportional to the video lengths. The rental price charged by the VP is usually proportional to the video length, which in turn is approximately proportional to the video size. Another motivating example is that for videos with similar lengths and compression ratios, the output video sizes are then determined by different spatial resolutions (e.g., the size of a movie with 1080p resolution is about 2.25 times of the size of that movie with 720p resolution). When the VP charges a rental price according to the video resolution provided, the rental price is also correlated to the video sizes. For the sake of simplification and without of loss of generality, we assume a proportional relationship between the rental price and the video size. Therefore, the overall rental cost of the MNO paid for renting the cached video copies is given by:

$$C_{MNO}(\delta, \boldsymbol{X}) = \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} s_j \cdot \delta \cdot x_{i,j}. \tag{8}$$

By subtracting the rental cost from the revenue gained by local caching, we can define for the MNO the utility function $U_{MNO}(\delta, \boldsymbol{X})$ in terms of the overall profit, as follows:

$$\begin{aligned} &U_{MNO}(\delta, \boldsymbol{X}) \\ &\quad = R_{MNO}(\boldsymbol{X}) - C_{MNO}(\delta, \boldsymbol{X}) \\ &\quad = \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} [P_j \lambda_i \cdot (\bar{R}_{i,j}^{save} + \bar{R}_{i,j}^{share}) - s_j \cdot \delta \cdot x_{i,j}]. \end{aligned} \tag{9}$$

On the VP side, the overall profit consists of three components: the revenue from leasing the cached video copies to the BSs owned by the MNO (which equals to the overall rental cost paid by the MNO), the revenue of user payment for downloading and watching requested videos, and the cost in terms of the shared user payment retained by the MNO with local caching, as follows.

$$\begin{aligned} U_{VP}(\delta, \boldsymbol{X}) = C_{MNO}(\delta, \boldsymbol{X}) + \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} P_j \lambda_i \cdot r_j \\ - \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} P_j \lambda_i \cdot \bar{R}_{i,j}^{share}. \end{aligned} \tag{10}$$

B. Stackelberg Game Formulation

Both the VP and the MNO are assumed to be selfish and intend to maximize their own revenues. From the above utility definition, however, it can be seen that the profit maximization of the VP and that of the MNO are two conflicting optimization objectives. On the one hand, the VP wishes to increase the unit rental price δ to earn more revenue from leasing videos. This would directly increase the rental cost and result in a lower profit for the MNO. On the other hand, following the law of demand, a higher unit rental price δ would reduce the amount of videos that the BSs are willing to rent for local caching, which may even lead to a lower overall profit for the VP itself. To reach the equilibrium between these two competing entities, game theory is an efficient approach to achieving their commonly accepted price and the optimal cache placement strategy. In the context of the hierarchical relationship between the VP and the MNO, Stackelberg game [6], [7], [9], [24] can be exploited as an extension of the noncooperative game for the resource competition between two groups of players, i.e., a leader and a follower.

We formulate the proposed video caching system into a single-leader single-follower Stackelberg game, where the VP acts as the leader while the MNO responds to the VP's action as the follower. More specifically, the VP first imposes a unit rental price δ and announces it to the MNO. Based on the unit rental price δ, the expected number of video requests submitted to the BSs and the average video request probability, the MNO then determines the amount of videos that it wants to rent for each BS and the optimal cache placement strategy $\boldsymbol{X} = \{x_{i,j}, \forall b_i \in \mathcal{B}, \forall f_j \in \mathcal{F}\}$ to maximize its utility/profit. Therefore, the proposed Stackelberg game is composed of the following two sub-problems: the profit maximization problem of the leader (VP) and the profit maximization problem of the follower (MNO).

1) VP's Profit Maximization Problem:

$$\textbf{P1:} \quad \max_{\delta} \quad U_{VP}(\delta, \boldsymbol{X}) = \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} [s_j \cdot \delta \cdot x_{i,j} + P_j \lambda_i \cdot r_j - P_j \lambda_i \cdot \bar{R}_{i,j}^{share}] \tag{11a}$$

$$\text{s.t.} \quad \delta \in [0, \delta_{\max}], \tag{11b}$$

$$x_{i,j} \in \{0, 1\}, \ \forall b_i \in \mathcal{B}, \ \forall f_j \in \mathcal{F}. \tag{11c}$$

The objective in Eq. (11a) is to maximize the VP's profit of renting the video files to the MNO, the constraint in Eq. (11b) states that the unit rental price does not exceed the maximum unit price set by the market, and the constraints in Eq. (11c) defines the cache placement strategy between the BSs and the video files as a matrix of binary decision variables.

2) MNO's Profit Maximization Problem:

$$\begin{aligned} \textbf{P2:} \quad &\max_{\boldsymbol{X}} U_{MNO}(\delta, \boldsymbol{X}) \\ &= \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} [P_j \lambda_i \cdot (\bar{R}_{i,j}^{save} + \bar{R}_{i,j}^{share}) - s_j \cdot \delta \cdot x_{i,j}] \\ &\triangleq \sum_{b_i \in \mathcal{B}} \sum_{f_j \in \mathcal{F}} V_{i,j}(\delta) \cdot x_{i,j} \end{aligned} \tag{12a}$$

$$\text{s.t.} \ Q_i = \sum_{f_j \in \mathcal{F}} s_j \cdot \mathbf{1}_{V_{i,j}(\delta) > 0}, \ \forall b_i \in \mathcal{B}, \tag{12b}$$

$$W_i = \min\{S_i, Q_i\}, \ \forall b_i \in \mathcal{B}, \tag{12c}$$

$$\sum_{f_j \in \mathcal{F}} x_{i,j} \cdot s_j \le W_i, \ \forall b_i \in \mathcal{B}, \tag{12d}$$

$$\delta \in [0, \delta_{\max}], \tag{12e}$$

$$x_{i,j} \in \{0, 1\}, \ \forall b_i \in \mathcal{B}, \ \forall f_j \in \mathcal{F}. \tag{12f}$$

The objective in Eq. (12a) is to maximize the MNO's profit as defined in Eq. (9), where we further define a notation $V_{i,j}(\delta)$

to denote the coefficient summation of the terms with respect to the specific decision variable $x_{i,j}$ in the total utility function $U_{MNO}(\delta, \boldsymbol{X})$. Its physical meaning is the profit of the MNO gained by caching video file f_j in BS b_i. The constraint in Eq. (12b) characterizes the price-demand relationship. For a given unit rental price δ, the MNO is willing to cache a video file f_j in its BS b_i only if by doing so a positive utility can be introduced (i.e., f_j is considered as a video file demanded for caching in b_i only if $V_{i,j}(\delta) > 0$). In Eq. (12b), $\mathbf{1}_{V_{i,j}(\delta)>0}$ is an indicator function the value of which equals to 1 if $V_{i,j}(\delta) > 0$ and 0 otherwise. The demand Q_i of BS b_i is then the summation of file sizes over the video files that could introduce a positive utility if cached. The constraint in Eq. (12c) defines the actual storage space W_i that BS b_i is willing to rent and able to cache due to the storage limit, as the minimum of the demand Q_i and the physical storage capacity S_i. The constraint in Eq. (12d) requires that the total size of cached video copies in BS b_i does not exceed its actual storage space W_i. The constraints in Eqs. (12e) and (12f) define the feasible regions for the price and cache placement decision variables, respectively.

3) Stackelberg Equilibrium: The target of the proposed Stackelberg game is to reach the Stackelberg Equilibrium (SE) point, from which neither the leader (VP) nor the follower (MNO) has any incentive to deviate [6], [7], [9]. If any entity deviates from this SE point, its own profit will be reduced. In the following, we define the SE point for the proposed Stackelberg game.

*Definition 1: Let δ^{opt} denote the optimal solution to the VP's profit maximization problem **P1** and $\boldsymbol{X}^{opt}$ denote the optimal solution to the MNO's profit maximization problem **P2** given the optimal unit rental price δ^{opt}. For any $(\delta, \boldsymbol{X})$ in the feasible region, if the following conditions hold:*

$$U_{VP}(\delta^{opt}, \boldsymbol{X}^{opt}) \geq U_{VP}(\delta, \boldsymbol{X}^{opt}), \tag{13}$$

$$U_{MNO}(\delta^{opt}, \boldsymbol{X}^{opt}) \geq U_{MNO}(\delta^{opt}, \boldsymbol{X}), \tag{14}$$

then (δ^{opt}, X^{opt}) is the SE point of the proposed Stackelberg game.

V. Iterative Dynamic Programming Algorithm for Noncooperative Caching

In this section, we focus on the noncooperative caching case and accordingly develop the iterative dynamic programming algorithm to jointly obtain the optimal pricing for the VP and the corresponding optimal cache placement strategy for the MNO.

The general process to find an SE point of the proposed Stackelberg game is to first solve the follower's optimization problem **P2** for the best response function $\boldsymbol{X}^*(\delta)$ given a unit rental price δ, and then to solve the leader's optimization problem **P1** for the optimal unit rental price δ^* based on the obtained $\boldsymbol{X}^*(\delta)$. However, since the decision variables $x_{i,j}, \forall b_i \in \mathcal{B}, \forall f_j \in \mathcal{F}$ are binary integers, the MNO's profit maximization problem is an integer programming problem, which means that $\boldsymbol{X}^*(\delta)$ cannot be explicitly expressed as a closed-form function. To address this issue, we develop in the following an iterative dynamic programming algorithm to obtain the SE point of the proposed Stackelberg game.

A. Problem Decomposition and Equivalent Knapsack Problems

We first solve the follower's optimization problem **P2** (i.e. the MNO's profit maximization problem) for a given unit rental price δ. Since the BSs in the noncooperative caching case only serve the mobile users with either the locally cached video copies or a video file transmitted from the VP, the cache placement decision variable $[x_{i,j}|\forall f_j \in \mathcal{F}]$ for a BS b_i is not coupled in the objective function or the constraints of problem **P2** with the decision variable in any other BS. The optimization problem **P2** can therefore be equivalently decomposed into a set of I sub-problems, with each sub-problem **SUB-i** corresponding to the cache placement optimization problem of BS $b_i \in \mathcal{B}$, as follows:

$$\textbf{SUB-i:} \quad \max_{[x_{i,j}|\forall f_j \in \mathcal{F}]} U_{b_i}(\delta, [x_{i,j}|\forall f_j \in \mathcal{F}]) \tag{15a}$$

$$\triangleq \sum_{f_j \in \mathcal{F}} [P_j \lambda_i \cdot (\bar{R}^{save}_{i,j} + \bar{R}^{share}_{i,j}) - s_j \cdot \delta \cdot x_{i,j}]$$

$$\text{s.t. } Q_i = \sum_{f_j \in \mathcal{F}} s_j \cdot \mathbf{1}_{V_{i,j}(\delta)>0}, \tag{15b}$$

$$W_i = \min\{S_i, Q_i\}, \tag{15c}$$

$$\sum_{f_j \in \mathcal{F}} x_{i,j} \cdot s_j \leq W_i, \tag{15d}$$

$$x_{i,j} \in \{0, 1\}, \forall f_j \in \mathcal{F}, \tag{15e}$$

where the decision variable $[x_{i,j}|\forall f_j \in \mathcal{F}]$ is a J dimensional cache placement vector, the objective function $U_{b_i}(\delta, [x_{i,j}|\forall f_j \in \mathcal{F}])$ in Eq. (15a) is defined as the profit gained by the MNO from the local caching in BS b_i, and the constraints in Eqs. (15b)-(15e) have the same physical meaning as the constraints in Eqs. (12b)-(12d) and (12f). By integrating Eqs. (3) and (4) into Eq. (15a), this objective function can be further derived as:

$$U_{b_i}(\delta, [x_{i,j}|\forall f_j \in \mathcal{F}])$$

$$= \sum_{f_j \in \mathcal{F}} [P_j \lambda_i \cdot (\bar{R}^{save}_{i,j} + \bar{R}^{share}_{i,j}) - s_j \cdot \delta \cdot x_{i,j}] \tag{16a}$$

$$= \sum_{f_j \in \mathcal{F}} [P_j \lambda_i s_j (c^{bh} + \varsigma \cdot d_{vp,i}) + P_j \lambda_i \theta r_j - s_j \delta] \cdot x_{i,j}$$

$$\triangleq \sum_{f_j \in \mathcal{F}} V_{i,j}(\delta) \cdot x_{i,j}, \tag{16b}$$

where $V_{i,j}(\delta) \triangleq P_j \lambda_i s_j (c^{bh} + \varsigma d_{vp,i}) + P_j \lambda_i \theta r_j - s_j \delta$ represents the profit of the MNO gained by caching video file f_j in BS b_i in the noncooperative caching case and its value remains constant for a given unit rental price δ.

By replacing the optimization objective in Eq. (15a) with that in Eq. (16b) and given a unit rental price δ, the actual storage space W_i can be computed according to Eqs. (15b) and (15c). The sub-problem **SUB-i** in Eq. (15) can then be viewed as a 0-1 knapsack problem. More specifically, the video files $f_1, \ldots, f_J \in \mathcal{F}$ are considered as the J items with each item f_j having a weight s_j and a value $V_{i,j}(\delta)$, while the maximum weight capacity of the knapsack is W_i. The profit maximization problem of BS b_i then converts to the following equivalent 0-1 knapsack problem:

$$\max_{[x_{i,j}\in\{0,1\}|\forall f_j\in\mathcal{F}]} \sum_{f_j\in\mathcal{F}} V_{i,j}(\delta)\cdot x_{i,j} \tag{17a}$$

$$\text{s.t.} \sum_{f_j\in\mathcal{F}} x_{i,j}\cdot s_j \le W_i, \tag{17b}$$

which optimally selects from a set of items $\mathcal{F}$ with different weight s_j and value $V_{i,j}(\delta)$ the items that maximize the total value subject to the knapsack's maximum capacity constraint W_i.

B. Dynamic Programming Algorithm

To efficiently solve the knapsack problem in Eq. (17), we develop a dynamic programming (DP) algorithm in Algorithm 1. The core idea of the DP algorithm is to decompose the original knapsack problem into a set of smaller knapsack problems, and to find the relationship between the structure of the optimal solution to the original knapsack problem and those to the smaller knapsack problems. Following this idea, we construct a two dimensional array $U[0\ldots J, 0\ldots W_i]$ to store the solutions to these smaller knapsack problems, where each entry $U[j,w], \forall 0 \le j \le J, \forall 0 \le w \le W_i$ stores the optimal solution (in terms of the maximum total value) to a smaller knapsack problem with the knapsack's capacity constraint reducing to $\sum_{f_{j'}\in\{f_1,f_2,\ldots,f_j\}} x_{i,j'}\cdot s_{j'} \le w$. That is, the value of $U[j,w]$ represents the maximum total value of any subset of the video file set $\{f_1, f_2, \ldots, f_j\}$ with the sum of video file sizes not larger than w. It can be recursively calculated based on the optimal solutions to some smaller knapsack problems, as follows:

$$U[j,w] = \begin{cases} 0, & \text{if } j = w = 0; \\ U[j-1,w], & \text{if } s_j > w; \\ \max\{U[j-1,w],\ \ V_{i,j}(\delta) + U[j-1,w-s_j]\}, & \\ & \text{if } s_j \le w. \end{cases} \tag{18}$$

Eq. (18) generally determines if the video file f_j will be selected by and contribute to the optimal solution $U[j,w]$, which can be studied in the following three cases. 1) When $j=0$ and $w=0$, then the value of $U[j,w]$ is set to zero. 2) When $s_j > w$, video file f_j cannot be added since otherwise the maximum capacity constraint is violated, which indicates $U[j,w] = U[j-1,w]$. 3) When $s_j \le w$, video file f_j is added only if by adding this item to the optimal solution of $U[j-1,w-s_j]$, the value of $V_{i,j}(\delta) + U[j-1,w-s_j]$ is larger than $U[j-1,w]$. At this time, $U[j,w]$ is updated as $V_{i,j}(\delta) + U[j-1,w-s_j]$. Otherwise, video file f_j cannot be added and we have $U[j,w] = U[j-1,w]$. The computational complexity of the proposed dynamic programming (DP) algorithm in Algorithm 1 is $\mathcal{O}(JW_i)$.

Algorithm 1 Dynamic programming algorithm for the knapsack problem in Eq. (17)

Input: vector $\boldsymbol{V_i}(\delta) = [V_{i,1}(\delta), V_{i,2}(\delta), \ldots, V_{i,J}(\delta)]$ defined in Eq. (16b);
vector of video file sizes $\boldsymbol{s} = [s_1, s_2, \ldots, s_J]$;
and actual storage space W_i for caching in BS b_i.
Output: optimal cache placement vector $[x^*_{i,j}|\forall f_j \in \mathcal{F}]$;
and maximum value of $U^*_{b_i}(\delta, [x^*_{i,j}|\forall f_j \in \mathcal{F}])$.

function DP($\boldsymbol{V_i}(\delta)$, $\boldsymbol{s}$, W_i)
for $w = 0$ to W_i **do**
  $U[0,w] := 0$
end for
for $j = 1$ to J **do**
  $U[j,0] := 0$
  for $w = 1$ to W_i **do**
    if $s_j \le w$ **then**
      if $V_{i,j}(\delta) + U[j-1,w-s_j] > U[j-1,w]$ **then**
        $U[j,w] := V_{i,j}(\delta) + U[j-1,w-s_j]$
        $Flag[j,w] := 1$
      else
        $U[j,w] := U[j-1,w]$
        $Flag[j,w] := 0$
      end if
    else
      $U[j,w] := U[j-1,w]$
      $Flag[j,w] := 0$
    end if
  end for
end for

$K := W_i$
for $j = J$ downto 1 **do**
  if $Flag[j,K] == 1$ **then**
    $x^*_{i,j} := 1$
    $K := K - s_j$
  else
    $x^*_{i,j} := 0$
  end if
end for
return $[x^*_{i,j}|\forall f_j \in \mathcal{F}] = [x^*_{i,1}, x^*_{i,2}, \ldots, x^*_{i,J}]$ and $U^*_{b_i}(\delta, [x^*_{i,j}|\forall f_j \in \mathcal{F}]) = U[J, W_i]$
end function

We implement the iterations over $j = 1, 2, \ldots, J$ and $w = 1, 2, \ldots, W_i$ in Algorithm 1 to obtain the value of $U[j,w]$ by utilizing at each iteration the recursive relationship in Eq. (18). In addition, we introduce an auxiliary array $Flag[1\ldots J, 1\ldots W_i]$ to record which subset of video files is selected for the optimal solution. Specifically, an entry $Flag[j,w], \forall 1 \le j \le J, \forall 1 \le w \le W_i$ is set to 1 if video file f_j is selected for the optimal solution $U[j,w]$; and set to zero otherwise. Then the maximum profit of the original knapsack problem in Eq. (17) is obtained as $U^*_{b_i}(\delta, [x^*_{i,j}|\forall f_j \in \mathcal{F}]) = U[J, W_i]$, and the corresponding optimal cache placement vector $[x^*_{i,j}|\forall f_j \in \mathcal{F}]$ can be obtained

Algorithm 2 Iterative dynamic programming algorithm for the noncooperative caching case

function MNO_Game($\delta, \boldsymbol{X}$)
for each BS $b_i \in \mathcal{B}$ **do**
  for each video file $f_j \in \mathcal{F}$ **do**
    Compute $V_{i,j}(\delta) := P_j \lambda_i s_j (c^{bh} + \varsigma d_{vp,i}) + P_j \lambda_i \theta r_j - s_j \delta$ according to Eq. (16b)
  end for
  Compute $Q_i := \sum_{f_j \in \mathcal{F}} s_j \cdot \mathbf{1}_{V_{i,j}(\delta)>0}$ according to Eq. (15b)
  Compute $W_i := \min\{Q_i, S_i\}$ according to Eq. (15c)
  DP($\boldsymbol{V_i}(\delta)$, $\boldsymbol{s}$, W_i) according to Algorithm 1
end for
return the optimal cache placement matrix $\boldsymbol{X}|_\delta = [x^*_{i,j}|\forall b_i \in \mathcal{B},\ \forall f_j \in \mathcal{F}]$ and the maximum MNO's profit $U_{MNO}(\delta, \boldsymbol{X}|_\delta) = \sum_{b_i \in \mathcal{B}} U^*_{b_i}(\delta, [x^*_{i,j}|\forall f_j \in \mathcal{F}])$, for a given unit rental price δ
end function

function VP_Game($\delta, \boldsymbol{X}$)
Set $\delta := 0$
repeat
  MNO_Game($\delta, \boldsymbol{X}$)
  Compute the VP's profit $U_{VP}(\delta, \boldsymbol{X}|_\delta)$ according to Eq. (10)
  $\delta \leftarrow \delta + \Delta_\delta$
until $U_{VP}(\delta, \boldsymbol{X}|_\delta)$ achieves the maximum value with regard to δ, and the corresponding decision variables are denoted as δ^{opt} and $\boldsymbol{X}^{opt}$
return the optimal unit rental price δ^{opt} and the optimal cache placement matrix $\boldsymbol{X}^{opt}$
end function

based on a downward iteration over $j = J, \ldots, 2, 1$ for the auxiliary entry $Flag[j, w], \forall 1 \leq j \leq J, \forall 1 \leq w \leq W_i$.

C. Iterative Dynamic Programming Algorithm

Based on Algorithm 1, we then develop a DP based iterative algorithm as shown in Algorithm 2 to find the SE point of the proposed Stackelberg game. The proposed DP based iterative algorithm in Algorithm 2 involves iterative rounds of interactions between the VP and the MNO. As a leader, the VP first initializes the unit rental price δ to zero and starts the game. In the follower's game, the MNO responds with computing the demanded amount of the video files and the actual storage space for local caching according to Eqs. (15b) and (15c), respectively, and running the DP algorithm in Algorithm 1 to obtain the optimal cache placement matrix and the corresponding maximum profit given the unit rental price. After that, the VP computes its own profit based on the cache placement matrix obtained by the MNO, and then increases the unit rental price δ by a slight increment of $\Delta_\delta > 0$ and implements the above interactions iteratively. The computational complexity of the proposed DP based iterative algorithm in Algorithm 2 is $\mathcal{O}(IJW_{\max}\delta_{\max}/\Delta_\delta)$, where $W_{\max} \triangleq \max_{b_i \in \mathcal{B}} W_i$.

At the first several iterations, the VP's profit would increase with the increment of δ. This is because when the unit rental price δ is small, the actual storage space (actual amount of rented video files for local caching) is mainly limited by and equal to the storage capacity of the BSs, and thus the total profit of the VP through renting will be linearly increased with the increment of δ. This increasing trend, however, cannot last for long. The demand of the BSs for local caching would be decreasing after the unit rental price δ reaches a certain point, which in turn would decrease the total profit of the VP. Therefore, after a few iterations, the VP's profit starts to decrease due to the decreasing demand of the BSs. The VP has to find such a point δ^{opt}, by running the iterations until its profit achieves the maximum value with regard to δ, and the corresponding unit rental price δ^{opt} and cache placement matrix $\boldsymbol{X}^{opt}$ are the SE point of the proposed Stackelberg game for the noncooperative caching case. In Algorithm 2, the value of step size Δ_δ should be sufficiently small to guarantee that the optimal solution $(\delta^{opt}, \boldsymbol{X}^{opt})$ is close enough to the SE point [9]. Then, for the optimal unit rental price δ^{opt} and the optimal cache placement matrix $\boldsymbol{X}^{opt}$ returned by the iterative DP algorithm in Algorithm 2 with a sufficiently small value of Δ_δ, we have $U_{VP}(\delta^{opt}, \boldsymbol{X}^{opt}) \geq U_{VP}(\delta, \boldsymbol{X}|_\delta)$ since $U_{VP}(\delta^{opt}, \boldsymbol{X}^{opt})$ achieves the maximum value among $U_{VP}(\delta, \boldsymbol{X}|_\delta)$ with regard to δ, and $U_{VP}(\delta, \boldsymbol{X}|_\delta) \geq U_{VP}(\delta, \boldsymbol{X}^{opt})$ since $\boldsymbol{X}|_\delta$ is the optimal cache placement matrix solved by the DP algorithm for given δ. Therefore, we can verify that $U_{VP}(\delta^{opt}, \boldsymbol{X}^{opt}) \geq U_{VP}(\delta, \boldsymbol{X}^{opt})$. In addition, $\boldsymbol{X}^{opt}$ is the optimal cache placement matrix solved by the DP algorithm when $\delta = \delta^{opt}$, thus we have $U_{MNO}(\delta^{opt}, \boldsymbol{X}^{opt}) \geq U_{MNO}(\delta^{opt}, \boldsymbol{X})$. Therefore, both the conditions in Eqs. (13) and (14) hold and the solution $(\delta^{opt}, \boldsymbol{X}^{opt})$ obtained by the proposed algorithm converges to the SE point. The convergence to the SE point of the proposed iterative DP algorithm in Algorithm 2 will also be justified by the experimental results shown in Fig. 3 in Section VII-A.

VI. Optimization Algorithm for Distributed Cooperative Caching

In this section, we focus on the distributed cooperative caching case and accordingly develop the iterative gradient ascent algorithm to find the SE point of the proposed Stackelberg game.

A. Challenges

Compared to the noncooperative caching case, the optimization algorithm design for the distributed cooperative caching is more complicated. The fundamental technical challenges introduced by the coordination between BSs can be explained as follows. If we integrate the expression of the notations $\bar{R}^{save}_{i,j}$ and $\bar{R}^{share}_{i,j}$ in Eqs. (5) and (6) into the optimization objective in Eq. (12a) and after derivation, we can see that the notation $V_{i,j}(\delta)$ includes not only the achieved revenue for serving the users of BS b_i by caching video file f_j in b_i, but also the achieved revenue for serving the users of the adjacent BSs to b_i by caching video file f_j in b_i. Therefore, the MNO's profit maximization problem in Eq. (12) cannot be decoupled and decomposed into the sub-optimization problems of its BSs, since the cache placement decision of a BS b_i is correlated with those of its adjacent BSs in $\mathcal{N}(b_i)$. In addition, the proposed dynamic programming algorithm in Algorithm 1 is no longer feasible, since the value of $V_{i,j}(\delta)$ is correlated with all the values of $V_{i',j}(\delta), \forall b_i' \in \mathcal{N}(b_i)$. In fact, following the proof of Proposition 1 in [5], we can similarly prove that in the noncooperative caching case the MNO's profit maximization problem in Eq. (12) for a given unit rental price δ is equivalent to a submodular maximization problem with a set of knapsack constraints[1]. In theory, it requires exponential computational complexity (with respect to I, J, and S_i) to obtain the optimal solution and usually is approximately solved in practice by the greedy algorithm with a sub-optimal solution to problem **P2**.

To address the above challenges, in the following, we introduce the split cache strategy designed for the cooperative caching system, and reformulate the MNO's profit maximization problem based on this strategy with the homogeneous assumptions on cache storage capacities and content sizes. We then extend the investigation for practical scenarios with heterogeneous cache storage capacities of BSs and heterogeneous file sizes of videos.

B. Split Cache Strategy for Distributed Cooperative Caching

The noncooperative cache placement strategy in Section V is to individually allocate the popular video files in the local cache of each BS by solving its own profit maximization problem, which is an extreme strategy in cooperative caching and might give rise to heavy content duplications over the network of MNO. Another extreme cache placement strategy is to allocate the popular video files over the BSs by avoiding duplications. These two strategies are either too greedy or too unselfish, resulting a poor caching performance. To strike the balance between greediness and full cooperation, the split cache strategy that locates somewhere in between these two extreme cases is proposed in [10], [25] for the distributed cooperative caching in small-cell networks. In addition, it is theoretically proved and numerically analyzed in [10] as the optimal cache placement strategy for networks with the homogeneous assumptions on content demands, cache storage capacities and content sizes.

Therefore, we adopt here this split cache strategy to divide the actual storage space of the cache into a duplicate partition and a unique partition. Specifically, the duplicate partition takes up a proportion ρ of the actual storage space and is used by every BS for caching the same most popular video files. The rest $1-\rho$ portion is allocated as the unique partition for disjointly caching unique subsets of the less popular video files in different BSs, such that the diversity of cached content is enhanced.

C. Problem Reformulation for MNO Profit Maximization

As illustrated in Fig. 2, we assume that each BS $b_i \in \mathcal{B}$ has the same storage capacity $S_i = S_0$, and the same demand

[1]The detailed proof is omitted here due to the space limit.

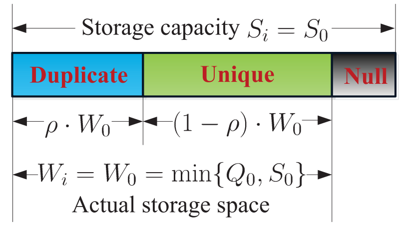

Fig. 2. Storage partitioning for the local cache of BS b_i based on the split cache strategy.

$Q_i = Q_0$ for a given unit rental price δ according to Eq. (12b). Therefore, the actual storage space used for caching can be obtained as $W_i = W_0 = \min\{Q_0, S_0\}$, within which a proportion of ρ is used for the duplicate partition while the rest for the unique partition. We further assume that each video file $f_j \in \mathcal{F}$ has the same file size $s_j = s_0$ and the same price $r_j = r_0$ charged by the VP from the user. The split cache strategy in [10], [25] then populates the duplicate partition of every BS with the N_{du} most popular video files $\mathcal{F}_{du} = \{f_1 \sim f_{N_{du}}\}$, where $N_{du} = \lfloor \rho \cdot W_0/s_0 \rfloor$ and the notation $\lfloor\cdot\rfloor$ denotes the floor function. On the other hand, a total number $N_{un} = I \cdot \lfloor (1-\rho) \cdot W_0/s_0 \rfloor$ of the unique video files will be uniformly distributed over the unique partitions of all the BSs $b_1 \sim b_I$, corresponding to the less popular video files $\mathcal{F}_{un} = \{f_{N_{du}+1} \sim f_{N_{du}+N_{un}}\}$. Therefore, based on Eq. (1), the probability that a requested video is cached in the duplicate partition of each BS is given by:

$$P^{du} = \sum_{f_j \in \mathcal{F}_{du}} P_j \approx \int_{j=1}^{N_{du}} \frac{\Omega}{j^\alpha} dj = \Omega \frac{N_{du}^{(1-\alpha)} - 1}{1-\alpha} \quad (19a)$$

$$= \frac{N_{du}^{(1-\alpha)} - 1}{J^{(1-\alpha)} - 1} \approx \frac{(\rho \cdot \frac{W_0}{s_0})^{(1-\alpha)} - 1}{J^{(1-\alpha)} - 1}, \quad (19b)$$

where $\Omega = 1/\sum_{j=1}^{J} \frac{1}{j^\alpha} \approx 1/(\int_{j=1}^{J} \frac{1}{j^\alpha} dj) = \frac{1-\alpha}{J^{(1-\alpha)}-1}$ and we have the approximation in Eq. (19b) by releasing the floor operation in N_{du}. The probability that a requested video is cached in the unique partitions of all the BSs can be similarly derived as:

$$P^{un} = \sum_{f_j \in \mathcal{F}_{un}} P_j = \sum_{j=1}^{N_{du}+N_{un}} P_j - \sum_{j=1}^{N_{du}} P_j \quad (20a)$$

$$= \frac{(N_{du} + N_{un})^{(1-\alpha)} - N_{du}^{(1-\alpha)}}{J^{(1-\alpha)} - 1} \quad (20b)$$

$$\approx \frac{\{(\rho \cdot \frac{W_0}{s_0}) + [I \cdot (1-\rho) \cdot \frac{W_0}{s_0}]\}^{(1-\alpha)} - (\rho \cdot \frac{W_0}{s_0})^{(1-\alpha)}}{J^{(1-\alpha)} - 1}, \quad (20c)$$

where the floor operation in N_{du} and N_{un} is similarly released to obtain the approximation in Eq. (20c).

Following the split cache strategy, the less popular video files $f_{N_{du}+1} \sim f_{N_{du}+N_{un}}$ will be uniformly distributed over the unique partitions of all the BSs. Therefore, the average local hit rate of each BS b_i, defined as the probability that the request from a user of b_i is served by the video files cached in b_i, can be expressed as:

$$P_i^L = P^{du} + \frac{P^{un}}{I} \tag{21}$$

The average adjacent BS hit rate of each BS b_i, defined as the probability that the request from a user of b_i is served by the video files cached in the adjacent BSs in $\mathcal{N}(b_i)$, is given by:

$$P_i^A = \frac{I-1}{I} P^{un}. \tag{22}$$

Both P_i^L and P_i^A can be written as functions of ρ.

Therefore, for a given unit rental price δ, the MNO's profit maximization problem in Eq. (12) can be reformulated with respect to the average local hit rate and the average adjacent BS hit rate, as follows:

$$\begin{aligned} \textbf{P3:} \max_{\rho\in[0,1]} U_{MNO}(\rho) \triangleq \sum_{b_i\in\mathcal{B}} \Bigg\{ & \lambda_i s_0 \ P_i^L \cdot (c^{bh} + \varsigma \cdot d_{vp,i}) \\ & + \lambda_i s_0 P_i^A \cdot \Bigg[(c^{bh} + \varsigma \cdot d_{vp,i}) - (c^{co} + \frac{\varsigma}{|\mathcal{N}(b_i)|} \cdot \\ & \sum_{n=1}^{|\mathcal{N}(b_i)|} d_{(n)_i,i}) \Bigg] \\ & + \lambda_i (P_i^L + P_i^A) \cdot \theta r_0 - s_0 \cdot \delta \cdot W_i \Bigg\}. \end{aligned} \tag{23}$$

In the reformulated optimization problem in Eq. (23), the decision variable changes to the proportion parameter ρ in the split cache strategy, and the optimization objective is expressed as a function of ρ. More specifically, in the objective function $U_{MNO}(\rho)$, the first term denotes the average revenue saved for sending requested videos to users of b_i by the locally cached video files in b_i, the second terms represents the average revenue saved for sending the requested videos to users of b_i by the remotely cached video files in the adjacent BSs of b_i, where $\frac{1}{|\mathcal{N}(b_i)|}\sum_{n=1}^{|\mathcal{N}(b_i)|} d_{(n)_i,i}$ is the average transmission delay between b_i and its adjacent BSs, the third term denotes the shared user payment achieved by serving the users with the cached video copy either in b_i or from its adjacent BSs, and the last term is the average rental cost in accordance with the actual storage space $W_i = W_0$ of b_i used for video caching, where the demand $Q_i = Q_0$ for a given unit rental price δ is obtained according to the constraint in Eq. (12b) and the actual storage space used for caching can be obtained as $W_i = W_0 = \min\{Q_0, S_0\}$ according to the constraint in Eq. (12c). In addition, the cache storage constraint in Eq. (12d) is satisfied when determining the number $N_{du} = \lfloor \rho \cdot W_0 / s_0 \rfloor$ of duplicated files and the number $N_{un} = I \cdot \lfloor (1-\rho) \cdot W_0 / s_0 \rfloor$ of unique files.

Proposition 1: The objective function $U_{MNO}(\rho)$ in Eq. (23) is a concave function over the feasible region of the decision variable $\rho \in [0, 1]$.

Proof: In general, the concavity of function $U_{MNO}(\rho)$ can be proved by showing that the second-order derivative $d^2\ U_{MNO}(\rho)/d\rho^2 \leq 0, \forall \rho \in [0, 1]$. For the details, please refer to Appendix IX. ■

D. Iterative Gradient Ascent Algorithm

Based on the proof of concavity in Proposition 1, the MNO's profit maximization problem can be efficiently solved by the gradient ascent algorithm. Then, similar to the procedure in Algorithm 2, we develop an iterative gradient ascent algorithm as shown in Algorithm 3 to find the SE point of the proposed Stackelberg game for the distributed cooperative caching case. In Algorithm 3, the VP's game and its interaction with the MNO are the same as those in Algorithm 2. The difference appears in the MNO's game that targets to obtain the optimal cache placement matrix and maximum MNO's profit for a given unit rental price δ. In the MNO's game of the distributed cooperative case, we first use the gradient ascent algorithm with a step-size of $\Delta_\rho > 0$ to find the optimal proportion parameter $\rho^*|_\delta$ for a given δ. Then, we populate the duplicate partition of every BS with the N_{du} most popular video files $f_1 \sim f_{N_{du}}$, and uniformly place the less popular video files $f_{N_{du}+1} \sim f_{N_{du}+N_{un}}$ over the unique partitions of all the BSs $b_1 \sim b_I$. The computational complexity of the proposed iterative gradient ascent algorithm in Algorithm 3 is $\mathcal{O}((\epsilon^{-2}+J)\delta_{\max}/\Delta_\delta)$, where the computational complexity of each gradient ascent algorithm in the MNO's game is $\mathcal{O}(\epsilon^{-2})$ according to [26].

Similarly as Algorithm 2, at the first several iterations when the unit rental price δ is small, the VP's profit would increase with the increment of δ. But after a few iterations, the VP's profit starts to decrease due to the decreasing demand of the BSs according to the price-demand relationship. The VP has to find such a point δ^{opt}, by running the iterations until its profit achieves the maximum value with regard to δ, and the corresponding unit rental price δ^{opt} and cache placement matrix $\boldsymbol{X}^{opt}$ are the SE point of the proposed Stackelberg game for the distributive cooperative caching case. The convergence to the SE point of the proposed iterative gradient ascent algorithm in Algorithm 3 will also be justified by the experimental results shown in Section VII-B.

E. Extension to Scenarios With Heterogeneous File Sizes and Cache Capacities

We now extend the analysis to a more practical scenario where the video files have different sizes and the cache storage capacities of BSs are also different. In this situation, the video files to be cached in the duplicate and unique partitions given the proportion parameter ρ are different than those in the homogeneous case. Specifically, the set of cached video files in the duplicate partition of every BS is obtained as:

$$\mathcal{F}_{du} = \arg\max_{\Psi\subseteq\mathcal{F}} \Bigg\{ \sum_{f_j\in\Psi} P_j \cdot s_j \Bigg| \sum_{f_j\in\Psi} s_j \leq \rho \cdot W_0 \Bigg\}, \tag{24}$$

where $W_0 = \min\{W_i | b_i \in \mathcal{B}\}$ and the set of video files that are uniformly distributed in the unique partitions of all the BSs is given by:

$$\begin{aligned} \mathcal{F}_{un} = \arg \max_{\Phi\subseteq(\mathcal{F}\setminus\mathcal{F}_{du})} \Bigg\{ & \sum_{f_j\in\Phi} P_j \cdot s_j \Bigg| \\ & \sum_{f_j\in\Phi} s_j \leq \sum_{b_i\in\mathcal{B}} W_i - (1-\rho) \cdot W_0 \Bigg\}, \end{aligned} \tag{25}$$

Algorithm 3 Iterative gradient ascent algorithm for the distributed cooperative caching case

1: **function** MNO_Game($\delta, \boldsymbol{X}$)
2: Set $\rho := 0$
3: Set error threshold ϵ to a small positive value close to 0 (e.g., $\epsilon = 0.1, 0.01, \ldots$)
4: Compute Q_0 according to Eq. (12b)
5: Compute $W_0 := \min\{Q_0, S_0\}$ according to Eq. (12c)
6: Set $U_{MNO} := U_{MNO}(\rho)$
7: **repeat**
8: $\quad \rho \leftarrow \rho + \Delta_\rho \cdot \frac{dU_{MNO}(\rho)}{d\rho}$
9: $\quad \Delta_{U_{MNO}} \leftarrow U_{MNO}(\rho) - U_{MNO}$
10: $\quad$ Set $U_{MNO} := U_{MNO}(\rho)$
11: **until** $|\frac{dU_{MNO}(\rho)}{d\rho}| \leq \epsilon$
12: Let $\rho^*|_\delta := \rho$
13: Compute $N_{du} := \lfloor \rho^*|_\delta \cdot W_0/s_0 \rfloor$
14: Compute $N_{un} := I \cdot \lfloor (1 - \rho^*|_\delta) \cdot W_0/s_0 \rfloor$
15: Set $\boldsymbol{X}|_\delta = [x^*_{i,j} | \forall b_i \in \mathcal{B},\ \forall f_j \in \mathcal{F}]$ as zero matrix.
16: Set $x^*_{i,j} := 1, \forall b_i \in \mathcal{B}, \forall f_j \in \mathcal{F}_{du} = \{f_1 \sim f_{N_{du}}\}$
17: **for** each video $f_j \in \mathcal{F}_{un} = \{f_{N_{du}+1} \sim f_{N_{du}+N_{un}}\}$ **do**
18: $\quad r \leftarrow (j \mod I) + 1$
19: $\quad$ Set $x^*_{r,j} := 1$
20: **end for**
21: **return** the optimal cache placement matrix $\boldsymbol{X}|_\delta$ and the maximum MNO's profit $U_{MNO}(\delta, \boldsymbol{X}|_\delta) = U_{MNO}(\rho^*|_\delta)$, for a given unit rental price δ
22: **end function**
23:
24: **function** VP_Game($\delta, \boldsymbol{X}$)
25: Set $\delta := 0$
26: **repeat**
27: $\quad$ MNO_Game($\delta, \boldsymbol{X}$)
28: $\quad$ Compute the VP's profit $U_{VP}(\delta, \boldsymbol{X}|_\delta)$ according to Eq. (10)
29: $\quad \delta \leftarrow \delta + \Delta_\delta$
30: **until** $U_{VP}(\delta, \boldsymbol{X}|_\delta)$ achieves the maximum value with regard to δ, and the corresponding decision variables are denoted as δ^{opt} and $\boldsymbol{X}^{opt}$
31: **return** the optimal unit rental price δ^{opt} and the optimal cache placement matrix $\boldsymbol{X}^{opt}$
32: **end function**

Then, the duplicate partition and unique caching probabilities in Eqs. (19a) and (20a) are normalized with respect to the file sizes, and rewritten as:

$$P_N^{du} = \sum_{f_j \in \mathcal{F}_{du}} P_j \cdot \frac{s_j}{s_0}, \text{ and } P_N^{un} = \sum_{f_j \in \mathcal{F}_{un}} P_j \cdot \frac{s_j}{s_0}, \quad (26)$$

where $s_0 = \sum_{f_j \in \mathcal{F}} s_j / J$ is the average file size. Based on Eqs. (21) and (22), we have the average local and adjacent BS hit rates, and plug them into Eq. (23) to obtain the MNO's profit maximization problem for the scenarios with heterogeneous video file sizes and cache storage capacities. For the heterogeneous case, we can still implement Algorithm 3 to find the SE point of the Stackelberg game between the VP and the MNO, but with a slight modification on the update step of the proportion parameter in Line 7 as $\rho \leftarrow \rho + \Delta_\rho$. That is, the previous gradient ascent algorithm used for the homogeneous case is modified here as a constant step-size search algorithm to find the optimal proportion parameter $\rho^*|_\delta$ for a given δ.

VII. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms for both the noncooperative and distributed cooperative caching cases, and derive simple guidelines for effective pricing and cache placement in video caching systems under different simulation settings.

A. Noncooperative Caching Case

We consider a video caching system comprising a VP that provides $J = 100$ video files with the Zipf parameter $\alpha = 0.5$, and an MNO with $I = 5$ BSs. The file sizes of the $J = 100$ videos at the VP are randomly selected from $\{s_0,\ 2s_0,\ 3s_0,\ 4s_0,\ 5s_0\}$, where $s_0 = 1$ is set as the normalized unit file size. For each BS $b_i \in \mathcal{B}$, the number of video download requests submitted from the users is assumed as $\lambda_i = 200$. For the backhaul links between the VP and BSs, we set the unit backhaul transmission cost $c^{bh} = 1$ dollars/unit size, the backhaul transmission delay $d_{vp,i} = 2$ s, and the weight constant $\varsigma = 0.1$. The revenue sharing percentage is set as $\theta = 0.45$ as in [22]. For the video price r_j paid by users, we assume that it depends on the video popularity, where the VP charges 0.5, 0.4 and 0.3 dollar/unit size from the users for video files $f_1 \sim f_{10}$, $f_{11} \sim f_{50}$ and $f_{51} \sim f_{100}$, respectively.

1) Impact of System Parameters: We first evaluate and compare the performance of the proposed iterative DP appeared in Algorithm 2 under various parameter settings, in order to gain a further insight into the impact of different system parameters. We adopt the aforementioned simulation settings, unless stated otherwise. Fig. 3 illustrates the impact of the unit rental price δ on VP's profit U_{VP} and MNO's profit U_{MNO}, and on the actual storage space W_i allocated for each BS b_i, by varying the parameter setting of the BS storage capacity S_i, the Zipf parameter α, and the unit backhaul transmission cost c^{bh}, respectively.

For different parameter settings, we can conclude some general observations, as follows. If we focus on any specific parameter setting, the actual storage space W_i for local caching keeps constant and equals to the storage capacity S_i when δ is small, and becomes a downward-sloping line after δ exceeds a certain value. This actual demand versus price curve empirically justifies the theoretical analysis of the law of demand in Section III-D, which claims that when other factors are kept equal, the demanded quantity of a good falls when the price rises. In accordance with this actual demand versus price curve, the maximum MNO's profit given a unit rental price δ also decreases with δ, since a larger δ results in a higher rental cost and a lower saved backhaul transmission cost due to the decreasing demand for local caching. The VP's profit first increases linearly with increasing δ when δ is small and the actual storage space W_i for local caching is larger than but constrained by the storage capacity S_i. Then,

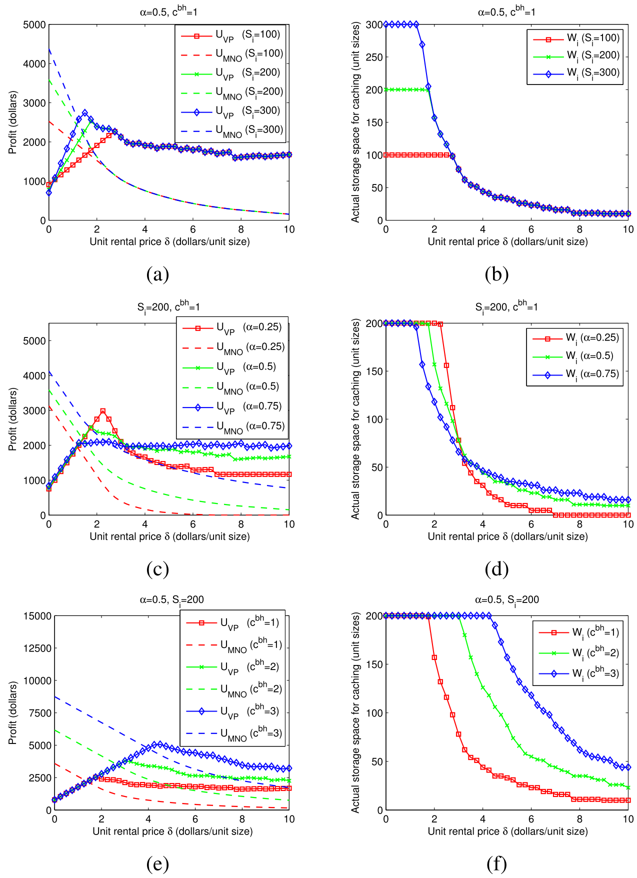

Fig. 3. Impact of the unit rental price δ on the profit of the VP and the MNO, and on the actual storage space allocated for BSs, with (a) and (b) different BS storage capacity S_i; (c) and (d) different Zipf parameter α; and (e) and (f) different unit backhaul transmission cost c^{bh}.

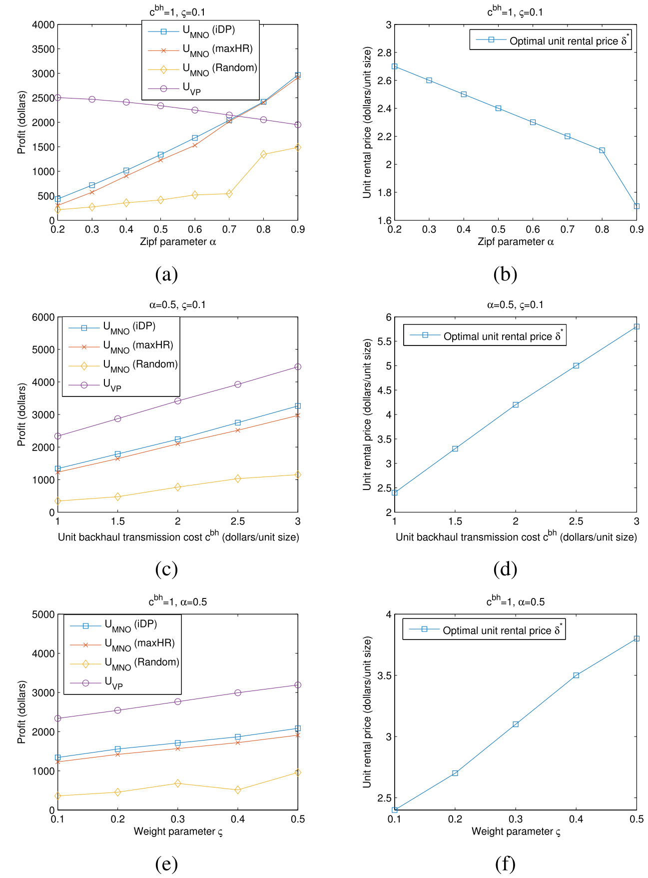

Fig. 4. (a) Maximum profit of the VP and the MNO and (b) optimal unit rental price vs. Zipf parameter; and (c) maximum profit of the VP and the MNO and (d) optimal unit rental price vs. unit backhaul transmission cost; and (e) maximum profit of the VP and the MNO and (f) optimal unit rental price vs. weight parameter; where the storage capacity of each BS is $S_i = 125$.

after a few iterations, the VP's profit starts to decrease when δ is larger than a certain point δ^{opt}, due to the decreasing demand of the BSs. This point δ^{opt}, together with the optimal cache placement matrix $\boldsymbol{X}^{opt}$ given $\delta = \delta^{opt}$, is thus the SE point of the proposed Stackelberg game. In other words, the convergence to the SE point of the proposed algorithm is justified in experiments.

More specifically, in Figs. 3(a) and 3(b), we let $\alpha = 0.5$ and $c^{bh} = 1$ dollars/unit size and evaluate the impact of δ on the algorithm's performance by setting S_i to 100, 200, and 300 unit sizes, respectively. It can be seen that as the storage capacity S_i increases, the optimal unit rental price δ^{opt} with regard to the SE point will decrease to allow a larger W_i. Although δ^{opt} decreases, the maximum profit of the VP and the MNO will both be increased due to the increment of W_i. Figs. 3(c) and 3(d) show the impact of δ on the algorithm's performance by fixing $c^{bh} = 1$ dollars/unit size and $S_i = 200$ unit sizes, and varying the value of Zipf parameter $\alpha = 0.25,\ 0.5,\ 0.75$. As α becomes larger, the video request probability is more concentrated on the popular video files, which results in a smoother actual demand versus price curve, and a smaller optimal unit rental price with a higher maximum profit for the MNO but a lower maximum profit for the VP. In Figs. 3(e) and 3(f), the impact of δ on the algorithm's performance is studied when $\alpha = 0.5$, $S_i = 200$ unit sizes, and c^{bh} changes from 1 to 3 dollars/unit size. We note that the actual storage space W_i starts to fall later when the unit backhaul transmission cost c^{bh} increases, which corresponds to a larger optimal unit rental price with a higher maximum profit for both the MNO and the VP.

2) Performance Comparison: We further compare the performance of the proposed iterative DP algorithm (iDP) in Algorithm 2 with two other caching schemes: *maximum hit rate (maxHR)*, where for a given unit rental price δ, each BS b_i selects to cache the video files that maximize the local cache hit rate until the demand is reached; and *random caching (Random)*, where for a given unit rental price δ, each BS b_i randomly selects the video files to cache until the demand is reached. In Fig. 4, we set storage capacity to $S_i = 125$ unit sizes, and compare the maximum profit of the VP and the MNO, and the optimal unit rental price in accordance with the SE point of each algorithm versus the Zipf parameter and the unit backhaul transmission cost, respectively. It should be noted that according to the law of demand and previous analysis on the demand versus price curve, for the same parameter setting, the optimal unit rental price after iterations to reach the SE point of all the three algorithms will be the same, as shown in Figs. 4(b) and 4(d). Similarly to the

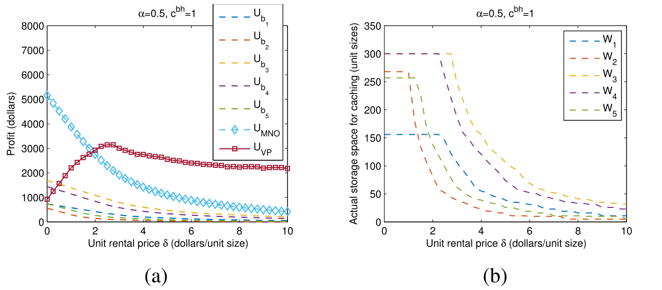

Fig. 5. Impact of the unit rental price δ on (a) the profit of the VP, the MNO and the BSs, and (b) the actual storage space allocated for BSs, under heterogeneous BS settings.

conclusion from Fig. 3, the optimal unit rental price of all the three algorithms decreases as the Zipf parameter increases, and increases as the unit backhaul transmission cost increases. Therefore, as shown in Figs. 4(a) and 4(c), the maximum profit of VP achieved by all the three algorithms is also the same, which represents a decreasing curve of α and an increasing curve of c^{bh}. In Figs. 4(e) and 4(f), we further study the impact of the weight parameter ς on the algorithms' performance. It can be seen that a larger value of ς results in a higher optimal unit rental price δ^{opt}, which corresponds to a higher maximum profit for both the MNO and the VP. For the MNO, the proposed algorithm can achieve a higher maximum profit than the other two schemes through the optimal cache placement strategy.

3) Performance Evaluation Under Heterogeneous BS Settings: Next, we study the performance of the proposed iterative DP algorithm in Algorithm 2 under the heterogeneous setting of the BSs, where we assume that the BSs have heterogeneous storage capacities $\boldsymbol{S} = [156,\ 268,\ 466,\ 362,\ 257]$ unit sizes (the capacities are randomly selected integers within the range $[100,\ 500]$), and different numbers of video requests $\boldsymbol{\lambda} = [231,\ 135,\ 385,\ 329,\ 182]$ (the numbers are randomly selected integers within the range $[100,\ 400]$). Fig. 5 shows the impact of δ on the performance of the proposed algorithm. As seen in Fig. 5(b), the actual demand versus price relationship of each BS represents a different curve since the parameter setting of S_i and λ_i becomes heterogeneous, which causes different cache placement decisions and thus different BS's profit versus unit rental price curves in Fig. 5(a). However, the overall profit of the VP and the MNO follows the same trend as in the homogeneous settings, and the SE point can still be reached by iteratively searching for the maximum value point of the VP's profit.

In Fig. 6, we compare the performance of the proposed algorithm with the other two caching schemes under the heterogeneous setting of the BSs. A similar trend as for the homogeneous setting in Fig. 4 can be observed: the optimal unit rental price and the maximum profit of VP achieved by all the three algorithms will decrease with the increment of Zipf parameter and increase with the increment of unit backhaul transmission cost, while the proposed algorithm achieves a higher maximum profit for MNO through the optimal cache placement strategy.

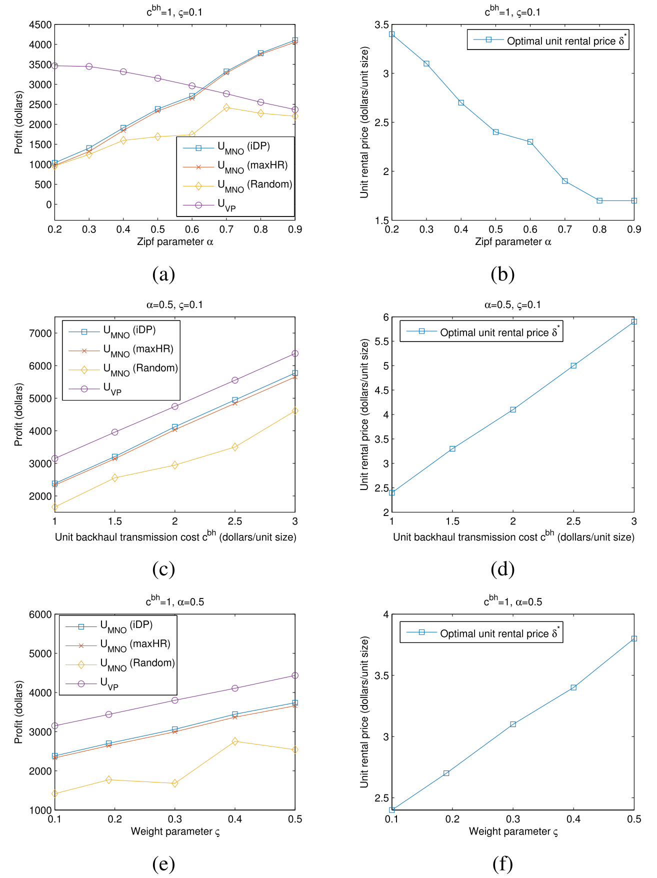

Fig. 6. (a) Maximum profit of the VP and the MNO and (b) optimal unit rental price vs. Zipf parameter; and (c) maximum profit of the VP and the MNO and (d) optimal unit rental price vs. unit backhaul transmission cost; and (e) maximum profit of the VP and the MNO and (f) optimal unit rental price vs. weight parameter; under heterogeneous BS settings.

B. Distributed Cooperative Caching Case

In the distributed cooperative caching case, the file size of all the $J = 100$ videos at the VP is first set as s_0, where $s_0 = 1$ unit size is the normalized unit file size. The video price charged by the VP from the users is set to 0.5 dollar/unit size for each video. We further assume that the transmission delay $d_{i,i'}$ between every two BSs b_i and $b_{i'}$ is randomly selected from the range $(0, 0.5)$ s. The remaining simulation settings are the same as in Section VII-A, unless stated otherwise.

1) Impact of System Parameters: We first evaluate and compare the performance of the proposed iterative gradient ascent algorithm in Algorithm 3 under various parameter settings, in order to gain a further insight into the impact of different system parameters. Fig. 7 presents the impact of the unit rental price δ on VP's profit U_{VP} and MNO's profit U_{MNO}, and on the actual storage space W_i allocated for each BS b_i, by varying the parameter setting of the BS storage capacity S_i, the Zipf parameter α, and the unit adjacent BS transmission cost c^{co}, respectively.

Similarly to the noncooperative caching case in Fig. 3, the general observation is that the law of demand in Section III-D can be justified by the actual demand versus price curves in Figs. 7(b), 7(d) and 7(f). In addition, the convergence to the SE point of the proposed iterative gradient

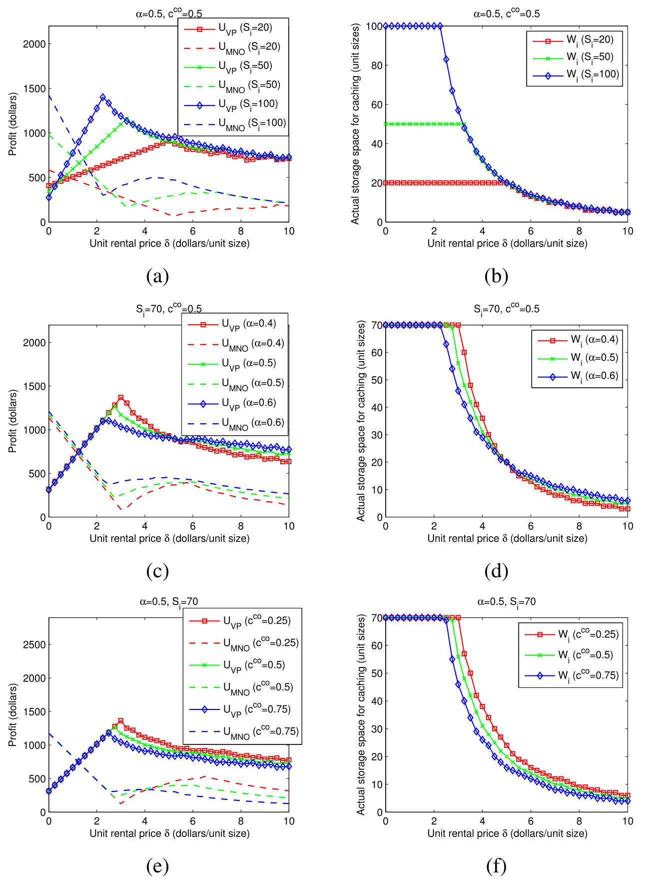

Fig. 7. Impact of the unit rental price δ on the profit of the VP and the MNO, and on the actual storage space allocated for BSs, with (a) and (b) different BS storage capacity S_i; (c) and (d) different Zipf parameter α; and (e) and (f) different unit adjacent BS transmission cost c^{co}.

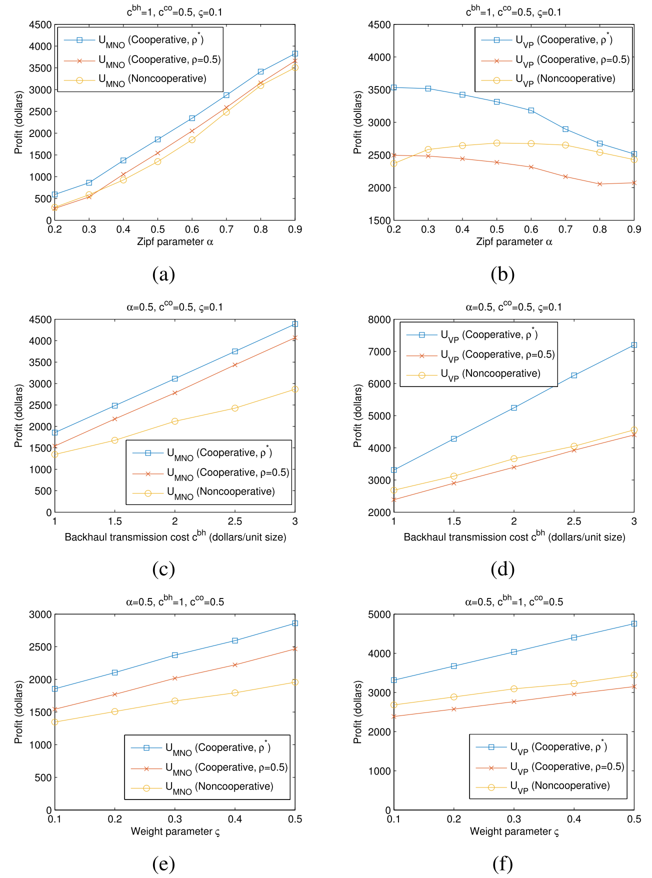

Fig. 8. Maximum profit of (a) the MNO and (b) the VP vs. Zipf parameter; and maximum profit of (c) the MNO and (d) the VP vs. unit backhaul transmission cost; and maximum profit of (e) the MNO and (f) the VP vs. weight parameter; under heterogeneous file size and caching capacity settings.

ascent algorithm is justified through the iterations of U_{VP} in Figs. 7(a), 7(c) and 7(e). More specifically, in Figs. 7(a) and 7(b), we let $\alpha = 0.5$ and $c^{co} = 0.5$ dollar/unit size and evaluate the impact of δ on the algorithm's performance by setting S_i to 20, 50, and 100 unit sizes, respectively. It can be seen that as the storage capacity S_i increases, the optimal unit rental price δ^{opt} with regard to the SE point will decrease to allow a larger W_i. Although δ^{opt} decreases, the maximum profit of the VP and the MNO will both be increased due to the increment of W_i. Figs. 7(c) and 7(d) show the impact of δ on the algorithm's performance by fixing $S_i = 70$ unit sizes and $c^{co} = 0.5$ dollars/unit size, and varying the value of Zipf parameter $\alpha = 0.4,\ 0.5,\ 0.6$. As α becomes larger, the video request probability is more concentrated on the popular video files, which results in a smoother actual demand versus price curve, and a smaller optimal unit rental price with a higher maximum profit for the MNO but a lower maximum profit for the VP. In Figs. 7(e) and 7(f), the impact of δ on the algorithm's performance is studied when $\alpha = 0.5$, $S_i = 70$ unit sizes, and c^{co} changes from 0.25 to 0.75 dollar/unit size. We note that the actual storage space W_i starts to fall earlier when the unit adjacent BS transmission cost c^{bh} increases, which corresponds to a smaller optimal unit rental price with a higher maximum profit for the MNO but a lower maximum profit for the VP.

2) Performance Evaluation Under Heterogeneous File Size and Cache Capacity Settings: Then, we conduct simulations for the scenario with heterogeneous file sizes and cache storage capacities, by setting the file sizes of the $J = 100$ videos the same as in Section VII-A and by assuming that the BSs have heterogeneous storage capacities $\boldsymbol{S} = [215,\ 252,\ 185,\ 120,\ 282]$ unit sizes (the capacities are randomly selected integers within the range $[100,\ 300]$) and different numbers of video requests $\boldsymbol{\lambda} = [296,\ 202,\ 251,\ 222,\ 276]$ (the numbers are randomly selected integers within the range $[200,\ 300]$). Here, the proposed iterative gradient ascent algorithm (denoted by cooperative, ρ^*) in Algorithm 3 is compared to other two schemes: the previously proposed iterative DP algorithm (denoted by noncooperative) in Algorithm 2 for the noncooperative caching, and the split cache strategy for the cooperative caching with a constant partitioning parameter $\rho = 0.5$ (denoted by cooperative, $\rho = 0.5$). Through the comparison result shown in Fig. 8, it can be seen that the proposed algorithm in Algorithm 3 achieves a higher maximum profit for both the VP and the MNO than the previously proposed optimal caching scheme in Algorithm 2 for the noncooperative caching. Such a performance gain is mainly achieved by encouraging the cooperation between BSs to further save the unit backhaul transmission cost, which becomes more significant when the difference between c^{bh}

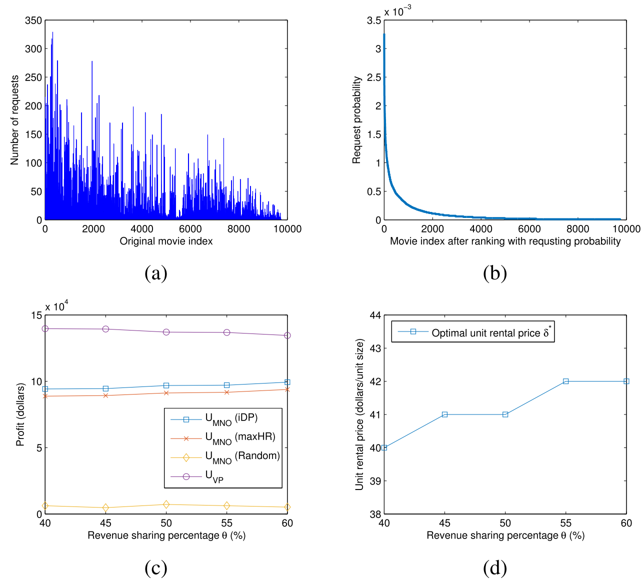

Fig. 9. (a) Number of video requests from users for different movies in MovieLens Latest Dataset (small), (b) request probability distribution after ranking with request probabilities, and (c) maximum profit of the VP and the MNO and (d) optimal unit rental price vs. revenue sharing percentage.

and c^{co} increases as shown in Figs. 8(c) and 8(d). In addition, the proposed algorithm in Algorithm 3 also outperforms the split cache strategy with a constant partitioning parameter $\rho = 0.5$, since the proposed algorithm is able to seek the optimal partitioning parameter ρ^* for any given unit rental price.

C. Performance Comparison on a Trace Dataset

Finally, we use the MovieLens Latest Dataset (small) [27] from MovieLens (an online recommender to gather research data on personalized recommendations) [28] as the trace dataset for performance comparison. This dataset contains 100835 ratings obtained from 610 users over 9742 movies, where each entry corresponds to a movie rating by a user and comprises an anonymous user ID, a movie ID, a rating made on a 5-star scale, and a timestamp. We adopt the similar approach as in [29], [30], and consider each movie rating as a video request from a user. To provide a better illustration of the dataset, Fig. 9(a) shows the number of video requests from users for different movies in the dataset, and Fig. 9(b) presents the request probability distribution of these movies after ranking with their request probabilities. Accordingly, we consider a video caching system comprising a VP that provides $J = 9742$ video files and an MNO with $I = 1$ BS. The spatial resolution of each video file is randomly selected from 360p, 480p and 720p. The corresponding file size is approximately assumed to be s_0, $2s_0$, and $5s_0$, where $S_0 = 1$ is set as the normalized unit file size. The storage capacity of the BS is set to 2000 unit sizes. For the video price r_j paid by users, we assume that it depends on the video popularity, where the VP charges 0.5, 0.4 and 0.3 dollar/unit size from the users for video files $f_1 \sim f_{200}$, $f_{201} \sim f_{1000}$ and $f_{1001} \sim f_{9742}$, respectively. The remaining simulation settings are the same as in Section VII-A, unless stated otherwise.

In Figs. 9(c) and 9(d), we compare the performance of the proposed iterative DP algorithm in Algorithm 2 with *maxHR* and *Random* schemes, by varying the revenue sharing percentage θ from 40% to 60%. The curves obtained for both the proposed iterative DP algorithm and the *maxHR* scheme shows that as θ increases, the MNO will retain more revenue from the shared user payment, which leads to an slightly increasing maximum profit for the MNO but a slightly decreasing maximum profit for the VP. It can also be seen that as θ becomes larger, the MNO is willing to pay a higher optimal rental price δ^* to the VP. In addition, through the optimal cache placement strategy, the proposed algorithm can achieve a higher maximum profit for the MNO under the same setting of θ than the other two schemes.

VIII. Conclusion

This paper studied a joint video pricing and cache placement problem for a video caching system that consists of a video provider (VP) and a mobile network operator (MNO) with a set of cache-enabled base stations (BSs). We have modeled the profit competition between the VP and the MNO as a Stackelberg game, and proposed a leader-follower optimization formulation to jointly maximize the profit of both entities by the optimal selection of video pricing for the VP and the optimal cache placement strategy for the BSs of the MNO, for both noncooperative and cooperative caching cases. For the noncooperative caching case, we have developed an iterative dynamic programming algorithm to efficiently find the SE point of the proposed Stackelberg game, while for the cooperative caching case, we have used the split cache strategy and developed an iterative gradient ascent algorithm to do so. Extensive simulations have been done under different system settings, empirically demonstrating the law of demand in economics and the convergence of the proposed algorithms to the SE point. Based on the analysis and simulation results, we further provided guidelines for the system design of the joint video pricing and cache placement. For the same system setting, the overall profit of both entities could be increased by allowing the coordination between the local caches of the BSs. In addition, the optimal video pricing as well as the optimal demand for local caching in the BSs were affected and thus could be adjusted by tuning different system parameters, such as the storage capacity, unit backhaul and adjacent BS transmission cost, distribution of video popularity. For future work, we plan to formally extend the proposed joint video pricing and cache placement optimization framework by exploiting the adaptive streaming technologies, such as dynamic adaptive streaming over HTTP (DASH).

IX. Proof of Proposition 1

For the sake of simplicity, we first let $c_1 = s_0 \cdot (c^{bh} + \varsigma \cdot d_{vp,i}) + \theta r_0$, $c_2 = s_0 \cdot (c^{co} + \frac{\varsigma}{|\mathcal{N}(b_i)|} \cdot \sum_{n=1}^{|\mathcal{N}(b_i)|} d_{(n)_i,i}) + \theta r_0$, and $c_3 = s_0 \cdot \delta \cdot W_i$, to represent some constant terms. Then, function $U_{MNO}(\rho)$ can be written as:

$$\begin{aligned} &U_{MNO}(\rho) \\ &\quad = \sum_{b_i \in \mathcal{B}} [\lambda_i c_1 \ P_i^L + \lambda_i (c_1 - c_2) P_i^A - c_3] && (27a) \\ &\quad = \sum_{b_i \in \mathcal{B}} [\lambda_i c_1 (P_i^L + P_i^A) - \lambda_i c_2 \ P_i^A - c_3] && (27b) \\ &\quad = \sum_{b_i \in \mathcal{B}} \left[\lambda_i c_1 (P^{du} + P^{un}) - \lambda_i c_2 \frac{I-1}{I} P^{un} - c_3 \right], && (27c) \end{aligned}$$

where Eq. (27c) is obtained by integrating the Eqs. (21) and (22). By further denoting $k_1 = \frac{W_0}{s_0}$ and $k_2 = \frac{1}{J^{(1-\alpha)}-1}$ in Eqs. (19) and (20), P^{du}, P^{un} and $(P^{du}+P^{un})$ can be written as:

$$P^{du} = k_2 (k_1 \rho)^{(1-\alpha)} - k_2, \quad (28)$$

$$P^{un} = k_2 [k_1 \rho + I k_1 (1-\rho)]^{(1-\alpha)} - k_2 (k_1 \rho)^{(1-\alpha)}, \quad (29)$$

$$P^{du} + P^{un} = k_2 [k_1 \rho + I k_1 (1-\rho)]^{(1-\alpha)} - k_2. \quad (30)$$

Therefore, the second-order derivative of $(P^{du} + P^{un})$ and P^{du} can be denoted and derived as:

$$\begin{aligned} \nabla &\triangleq \frac{d^2 (P^{du} + P^{un})}{d\rho^2} = -\alpha(1-\alpha) k_2 k_1^2 (1-I)^2 \cdot \\ &\quad [k_1 \rho + I k_1 (1-\rho)]^{(-\alpha-1)} \leq 0, && (31) \\ \nabla^{du} &\triangleq \frac{d^2 P^{du}}{d\rho^2} = -\alpha(1-\alpha) k_2 k_1^2 (k_1 \rho)^{(-\alpha-1)} \leq 0. && (32) \end{aligned}$$

Then, for the second-order derivative of function $U_{MNO}(\rho)$, we have:

$$\begin{aligned} &\frac{d^2 \ U_{MNO}(\rho)}{d\rho^2} \\ &\quad = \sum_{b_i \in \mathcal{B}} \left[\lambda_i c_1 \nabla - \lambda_i c_2 \frac{I-1}{I} (\nabla - \nabla^{du}) \right] && (33a) \\ &\quad = \sum_{b_i \in \mathcal{B}} \left[\lambda_i (c_1 - c_2 \frac{I-1}{I}) \nabla + \lambda_i c_2 \frac{I-1}{I} \nabla^{du} \right] && (33b) \\ &\quad \leq 0, && (33c) \end{aligned}$$

where the inequality in Eq. (33c) holds since backhaul transmission cost $s_0 \cdot (c^{bh} + \varsigma \cdot d_{vp,i})$ is larger than the adjacent BS transmission cost $s_0 \cdot (c^{co} + \frac{\varsigma}{|\mathcal{N}(b_i)|} \cdot \sum_{n=1}^{|\mathcal{N}(b_i)|} d_{(n)_i,i})$, which indicates that $c_1 > c_2$. Therefore, function $U_{MNO}(\rho)$ is a concave function, and Proposition 1 is proved.

REFERENCES

[1] *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*, Cisco, San Jose, CA, USA, 2017. [Online]. Available: https://goo.gl/kvPZPc

[2] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[3] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[4] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[5] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 965–984, Apr. 2018.

[6] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient video pricing and caching in heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8744–8751, Oct. 2016.

[7] J. Li, H. Chen, Y. Chen, Z. Lin, B. Vucetic, and L. Hanzo, "Pricing and resource allocation via game theory for a small-cell video caching system," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2115–2129, Aug. 2016.

[8] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[9] Z. Chen, Y. Liu, B. Zhou, and M. Tao, "Caching incentive design in wireless D2D networks: A Stackelberg game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[10] M. Taghizadeh, K. Micinski, C. Ofria, E. Torng, and S. Biswas, "Distributed cooperative caching in social wireless networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 6, pp. 1037–1053, Jun. 2013.

[11] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, May 2017.

[12] W. Li, S. M. A. Oteafy, and H. S. Hassanein, "Dynamic adaptive streaming over popularity-driven caching in information-centric networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5747–5752.

[13] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1914–1925, Dec. 2015.

[14] G. Gao, W. Zhang, Y. Wen, Z. Wang, and W. Zhu, "Towards cost-efficient video transcoding in media cloud: Insights learned from user viewing patterns," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1286–1296, Aug. 2015.

[15] H. Zhao, Q. Zheng, W. Zhang, B. Du, and H. Li, "A segment-based storage and transcoding trade-off strategy for multi-version VoD systems in the cloud," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 149–159, Jan. 2017.

[16] M. Hajimirsadeghi, N. B. Mandayam, and A. Reznik, "Joint caching and pricing strategies for popular content in information centric networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 654–667, Mar. 2017.

[17] K. Poularakis and L. Tassiulas, "On the complexity of optimal content placement in hierarchical caching networks," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2092–2103, Mar. 2016.

[18] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 126–134.

[19] (2018). *Economics Basics: Supply and Demand*. [Online]. Available: https://www.investopedia.com/university/economics/economics3.asp

[20] N. G. Mankiw, *Principles Microeconomics*, 5th ed. Boston, MA, USA: Cengage Learning, 2008.

[21] J. A. A. Van der Veen and V. Venugopal, "Using revenue sharing to create win–win in the video rental supply chain," *J. Oper. Res. Soc.*, vol. 56, no. 7, pp. 757–762, 2005.

[22] J. H. Mortimer, "Vertical contracts in the video rental industry," *Rev. Econ. Stud.*, vol. 75, no. 1, pp. 165–199, Jan. 2008.

[23] K. Ho, J. Ho, and J. H. Mortimer, "The use of full-line forcing contracts in the video rental industry," *Amer. Economic Rev.*, vol. 102, no. 2, pp. 686–719, Apr. 2012.

[24] X. Kang, R. Zhang, and M. Motani, "Price-based resource allocation for spectrum-sharing femtocell networks: A stackelberg game approach," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 538–549, Apr. 2012.

[25] Z. Chen, J. Lee, T. Q. S. Quek, and M. Kountouris, "Cooperative caching and transmission design in cluster-centric small cell networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3401–3415, May 2017.

[26] C. Cartis, N. I. M. Gould, and P. L. Toint, "On the complexity of steepest descent, Newton's and regularized newton's methods for nonconvex unconstrained optimization problems," *SIAM J. Optim.*, vol. 20, no. 6, pp. 2833–2852, 2010.

[27] *MovieLens Latest Datasets*. Accessed: Mar. 12, 2019. [Online]. Available: https://grouplens.org/datasets/movielens/latest/

[28] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, Jan. 2016.

[29] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1024–1036, Feb. 2017.

[30] N. Zhang, K. Zheng, and M. Tao, "Using grouped linear prediction and accelerated reinforcement learning for online content caching," in *Proc. IEEE Int. Conf. Commun. Workshops*, May 2018, pp. 1–6.

Junni Zou (M'07) received the M.S. and Ph.D. degrees in communication and information system from Shanghai University, Shanghai, China, in 2004 and 2006, respectively.

From 2006 to 2017, she was with the School of Communication and Information Engineering, Shanghai University, where she was a Full Professor. From 2011 to 2012, she was a visiting Professor with the Department of Electrical and Computer Engineering, University of California at San Diego (UCSD). Since 2017, she has been a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include multimedia networking and communications, distributed resource allocation, and machine learning. She has published over 90 international journal/conference papers on these topics. He received the Best 10% Paper Award at the IEEE VCIP 2016. She was granted the National Science Fund for Outstanding Young Scholar in 2016. She was a recipient of the Shanghai Young Rising-Star Scientist in 2011. She has been serving as an Associate Editor for *Digital Signal Processing* since 2019.

Chenglin Li (S'12–M'15) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007, 2009, and 2015, respectively.

From 2011 to 2013, he was a Visiting Ph.D. Student with the Electrical and Computer Engineering Department, University of Florida, Gainesville, FL, USA. From 2015 to 2017, he was a Post-Doctoral Research Fellow with the Signal Processing Laboratory (LTS4), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. From 2017 to 2018, he was a Senior Researcher with the Chair of Media Technology (LMT), Technical University of Munich (TUM), Munich, Germany, supported by the Hildegard Maier Research Fellowship of the Alexander von Humboldt Foundation for postdoctoral researchers. Since 2018, he has been an Associate Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. His main research interests include network oriented image/video processing and communication, network-based optimization for video sources, adaptive multimedia communication systems, and caching and edge computing for multimedia delivery. He was awarded the Microsoft Research Asia (MSRA) Fellow in 2011 and the Alexander von Humboldt Research Fellow in 2017. He received the IEEE VCIP Best 10% Paper Award in 2016.

Congcong Zhai received the B.S. degree in communication and information system from Shanghai Second Polytechnic University, Shanghai, China, in 2015, and the M.S. degree in communication and information system from Shanghai University, Shanghai, in 2018. His research interests include video networking and communications, resource allocation, and game theory.

Hongkai Xiong (M'01–SM'10) received the Ph.D. degree from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2003.

He was an Associate Professor from 2005 to 2011 and an Assistant Professor from 2003 to 2005. From 2007 to 2008, he was a Research Scholar with the Department of Electrical and Computer Engineering, Carnegie Mellon University (CMU), Pittsburgh, PA, USA. From 2011 to 2012, he was a Scientist with the Division of Biomedical Informatics, University of California at San Diego (UCSD), La Jolla, CA, USA. He is currently a Distinguished Professor with the Department of Electronic Engineering and the Department of Computer Science and Engineering, SJTU. He is also the Vice Dean of the Zhiyuan College, SJTU. Since then, he has been with the Department of Electronic Engineering, SJTU. He has published over 200 refereed journal and conference papers. He was the co-author of the Top 10% Paper in IEEE MMSP 2011, the Best Paper in IEEE BMSB 2013, the Best Student Paper in IEEE VCIP 2014, the Top 10% Paper in IEEE VCIP 2016. His research interests include multimedia signal processing, image and video coding, multimedia communication and networking, computer vision, biomedical informatics, and machine learning.

Dr. Xiong's research projects are funded by NSFC, QUALCOMM, Microsoft, and Intel. He received the New Century Excellent Talent Award from Ministry of Education of China in 2009, the SJTU SMC-A Excellent Young Faculty Awards in 2010 and 2013, the Shanghai Shu Guang Scholar Award in 2013, the Shanghai Youth Science and Technology Talent Award in 2014, the National Science Fund for Distinguished Young Scholar Award from Natural Science Foundation of China (NSFC) in 2014, the Yangtze River Scholar Distinguished Professor Award from the Ministry of Education of China in 2016, the Science and Technology Innovative Leader Talent Award in Ten Thousand Talents Program in 2017, the Shanghai Academic Research Leader Talent Award in 2017, the Baosteel Excellent Faculty Award in 2017, and the Shanghai Youth Science and Technology Distinguished Accomplishment Award in 2018. He has been granted the First Prize of the Shanghai Technology Innovation Award in 2011 and 2017, respectively, for research achievements on network-oriented video processing and dissemination. He is also an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT).

Eckehard Steinbach (M'96–SM'08–F'15) studied electrical engineering at the University of Karlsruhe, Germany, the University of Essex, U.K., and ESIEE Paris. He received the Ph.D. degree in engineering from the University of Erlangen-Nuremberg, Germany, in 1999. From 1994 to 2000, he was a member of the Research Staff of the Image Communication Group, University of Erlangen-Nuremberg. From 2000 to 2001, he was a Post-Doctoral Fellow with the Information Systems Laboratory, Stanford University. In 2002, he joined the Department of Electrical Engineering and Information Technology, Technical University of Munich, Germany, where he is currently a Full Professor of media technology. His current research interests are in the area of audio-visual-haptic information processing and communication, and networked and interactive multimedia systems.