Joint Optimization of Video Recommendation and Cooperative Edge Caching for Maximizing Profit

Haoqiu Luo[1], Youling Zeng[1], Yufan Shen[1], Yue Zeng[1], Liying Li[1], Qianpiao Ma[1], Peijin Cong[1], and Junlong Zhou[1,2]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, China

2026 IEEE/ACM International Symposium on Quality of Service (IWQoS) | 979-8-3195-1703-6/26/$31.00 ©2026 IEEE | DOI: 10.1109/IWQoS70441.2026.11661230

***Abstract*—The rapid growth of multi-version video-on-demand (VoD) services places increasing pressure on mobile networks. Edge caching offers a promising solution by bringing content closer to mobile users (MUs). However, practical edge caching systems face a fundamental challenge: MUs exhibit highly varied preferences across video contents and bitrates, whereas edge resources are limited, costly, and influenced by pricing decisions from mobile network operators (MNOs). In this paper, we study a novel, profit-oriented joint optimization problem that integrates personalized video recommendation, cooperative multi-version edge caching, soft cache hits, and economic incentives. We model the interaction between the MNO and the video provider (VP) as a Stackelberg game, which explicitly captures the coupling between pricing decisions, caching strategies, and various demands of MUs. We also show that the studied problem is NP-hard due to its non-convex structure and the presence of mixed discrete–continuous decision variables. To tackle this challenge, we propose a Backward-Induction Stackelberg framework, in which the MNO optimizes cache-leasing prices via an iterative distance-weighted strategy, while the VP solves its NP-hard joint decision problem through block coordinate descent combined with relaxation-and-rounding techniques. Extensive simulation results show that the proposed framework significantly outperforms state-of-the-art schemes, improving the MNO's revenue by up to 32.18% and the VP's revenue by up to 33.43%, while remaining robust under diverse system configurations.**

***Index Terms*—Edge Caching, Video Recommendation, Soft Hit, Stackelberg Game, Relaxation and Rounding**

I. Introduction

A. Background

In the era of the digital economy, video services have become the dominant component of global Internet traffic. Recent reports indicate that global monthly mobile data traffic reached 172 EB in early 2025 and is projected to grow to 430 EB by 2030, with video content accounting for over 74% of total traffic [1]. The rise of 5G and 6G-enabled interactive services, along with multi-version video-on-demand (VoD) systems, has imposed unprecedented pressure on mobile backbone networks [2]. To alleviate this pressure, edge caching has emerged as a key enabling technology by bringing popular content closer to end users, thereby reducing backhaul traffic and improving content delivery efficiency [3].

✉ Corresponding Authors: Peijin Cong and Junlong Zhou. Contact e-mail: 125106010723@njust.edu.cn.

979-8-3195-1703-6/26/$31.00 ©2026 IEEE

The edge caching ecosystem typically involves a mobile network operator (MNO) and a video provider (VP), each with distinct economic objectives. The MNO sets cache-leasing prices while the VP decides which video versions to cache and how to deliver them to mobile users (MUs) under limited edge resources. Under this leader–follower interaction, the VP must make recommendation, caching and delivery decisions based on the MNO's leasing price and the limited availability of edge resources. However, users exhibit highly diverse preferences for video contents and bitrates, whereas edge cache resources are often limited, costly, and allocated statically. This discrepancy leads to inefficient cache utilization, increased backhaul traffic, and lower profits for both MNO and VP.

To address the inefficiencies from heterogeneous multi-version demand under limited edge resources, prior work falls into two categories: cache-centric designs including cooperative caching and demand-aware enhancements including soft hits or recommendations. Additionally, economic models such as Stackelberg games and contract-theoretic frameworks have been introduced to incorporate pricing incentives into edge caching systems. However, these strands of work are often examined independently, motivating a systematic review and an integrated, profit-oriented solution, as discussed next.

B. Related Work and Motivation

1) Cache-Centric Designs: Cache-centric designs are a major area of research focused on improving hit ratio, reducing latency, or minimizing backhaul traffic through a variety of methods such as optimization-based, heuristic, and learning-driven methods [2], [4]–[8]. Cooperative caching further extends these efforts by facilitating content sharing among neighboring edge nodes, which improves storage utilization [3], [9], [10]. However, most cache-centric strategies neglect the economic relationship between MNOs and VPs, failing to capture how cache-leasing prices impact multi-version content placement. Although game-theoretic models have been introduced to tackle this issue [11]–[13], they often separate pricing from caching decisions or assume all content sizes are the same, limiting their applicability to realistic VoD systems.

2) Demand-Aware Enhancements: Beyond static caching, demand-aware enhancements aim to better align user requests with cached content through proactive recommendations and soft cache hit mechanisms. Existing recommendation models

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

often focus on maximizing click-through rates but tend to overlook cache states and version-dependent delivery costs [14]–[17]. Soft cache hit schemes exploit content or bitrate substitutability to mitigate the impact of cache misses [18], [19]. However, they often depend on simplified user acceptance models and ignore their interaction with recommendation and transcoding decisions. While pricing incentives have been incorporated into contract-theoretic and Stackelberg-based frameworks [11]–[13], [20]–[22], these approaches typically study demand shaping and caching in isolation, without jointly optimizing them in multi-version VoD environments.

3) Research Gap: In summary, existing studies do not provide a unified, profit-oriented framework that simultaneously optimizes personalized recommendation, cooperative caching, and multi-version delivery while considering explicit economic incentives. This gap motivates the need to develop a game-driven approach that aligns pricing decisions with demand shaping and cache utilization.

C. Our Approach

Inspired by the identified gap, this paper studies the joint optimization of personalized video recommendation and cooperative edge caching to maximize the profits of both MNO and VP. The interaction between these two parties is modeled as a Stackelberg game, which explicitly captures the coupling among user preference heterogeneity, cache-leasing prices, and multi-version content substitutability. We show that the resulting optimization problem is NP-hard due to its non-convex structure and the coupling of discrete caching decisions with continuous pricing variables. To solve this problem, we propose a Backward-Induction Stackelberg (BIS) scheme driven by economic incentives, which integrates iterative pricing search, block coordinate descent, and relaxation-and-rounding techniques. The main contributions are as follows.

- We formulate a profit-oriented joint optimization problem that integrates personalized recommendation, cooperative multi-version edge caching, soft cache hits, and economic incentives. We model the MNO-VP interaction as a Stackelberg game with provable computational complexity.
- We propose a hierarchical, game-driven optimization framework based on backward induction, in which the MNO (leader) determines the cache-leasing price via an iterative distance-weighted pricing strategy, and the VP (follower) efficiently solves its NP-hard joint optimization problem through block coordinate descent and relaxation-and-rounding techniques.
- Trace-driven experiments show significant performance improvements over state-of-the-art methods, increasing the MNO's revenue by up to 32.18% and the VP's revenue by up to 33.43%, while remaining robust across diverse system configurations.

II. System Overview and Problem Formulation

A. Network Model

As shown in Fig. 1, the system comprises a VP (e.g., iQIYI or YouTube), an MNO managing a series of small base stations (SBSs) with caching and transcoding capabilities, and multiple MUs who request videos via wireless links. Upon receiving the video, MUs pay the VP based on their quality of experience (QoE). To reduce latency, the VP leases cache space from the MNO. Both aim to maximize profit, motivating the joint optimization of MNO pricing and VP recommendation, caching, and delivery strategies.

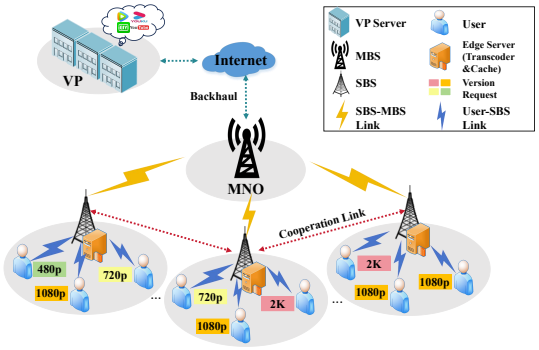


Fig. 1: Edge caching system for multi-version VoD service.

A video collection $\mathbf{V} = \{1, 2, \cdots, V\}$ stored in the VP contains V videos. Each video $v \in \mathbf{V}$ is categorized into M categories $\mathbf{M} = \{1, 2, \cdots, M\}$, and has Q bitrate versions $\mathbf{Q} = \{1, 2, \cdots, Q\}$ which are sorted in descending order of bitrate. Inspired by [23], each video v is described by category features $\mathbf{f}_v = \{f_{v,m}\}$. We denote the q-th bitrate version of v as (v, q). The size of video version (v, q) is $b_q L_v$, where b_q is the q-th bitrate and L_v is the duration of the video.

The MNO includes a macro base station (MBS) covering a set of SBSs, denoted as $\mathbf{S} = \{1, 2, \cdots, S\}$. Each SBS s is characterized by caching capacity c_s, CPU frequency f_s, and computation resource D_s. SBS s serves a set of directly connected MUs $\mathbf{B}_s$, while collaborating with N neighboring SBSs, $\mathbf{N}_s$, ranked by transmission rate. Under the cache-leasing contract, the MNO sets a basic cache-leasing price δ per unit capacity. SBS s adjusts this base price to its price δ_s using a distance-weighted pricing method [24]:

$$\delta_s = \frac{d_{M,s}}{d^{\text{avg}}}\delta, \tag{1}$$

where $d_{M,s}$ is its distance to the MNO and d^{avg} the average.

Let $\mathbf{U} = \{1, 2, \cdots, U\}$ denote the set of MUs served by the VP, where U is the total number of MUs. Each MU u accesses videos through its associated SBS s_u and pays the VP based on QoE. MU u has specific preferences, including category preference [23] $\mathbf{f}_u = \{f_{u,m}\}$ and bitrate preference $\mathbf{f}_u^{\text{b}} = \{f_{u,q}^{\text{b}}\}$. Combined with category preference $f_{u,m}$ and category feature $f_{v,m}$, the content matching degree [18] between MU u's preference and video v is defined as

$$a_{u,v}^{\text{pref}} = \frac{\sum_{m\in\mathbf{M}} f_{v,m} f_{u,m}}{\sqrt{\sum_{m\in\mathbf{M}} (f_{v,m})^2}\sqrt{\sum_{m\in\mathbf{M}} (f_{u,m})^2}}. \tag{2}$$

The degree can be normalized to a preference probability

$$p_{u,v}^{\text{pref}} = \frac{a_{u,v}^{\text{pref}}}{\sum_{i\in\mathbf{V}} a_{u,i}^{\text{pref}}}, \quad \forall u \in \mathbf{U}. \tag{3}$$

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

B. Recommendation and MU Request Model

The VP generates binary recommendations $r_{u,v} \in \{0,1\}$, where $r_{u,v}=1$ indicates that video v is recommended to MU u. Let $o_u \in [0,1]$ denote the probability that MU u accepts a recommendation video. The request probability [25] for video v from MU u is formulated as

$$p^{\mathrm{req}}_{u,v} = o_u \frac{r_{u,v} p^{\mathrm{pref}}_{u,\nu}}{\sum_{i\in\mathbf{R}_u} p^{\mathrm{pref}}_{u,i}} + (1-o_u)\frac{(1-r_{u,v})p^{\mathrm{pref}}_{u,v}}{\sum_{i\in\mathbf{V}\setminus\mathbf{R}_u} p^{\mathrm{pref}}_{u,i}}, \quad (4)$$

where $\mathbf{R}_u$ is MU u's personalized recommendation list. Similar to [18], by incorporating the MU u's bitrate feature f^b_u, the MU u's request probability for (v,q) is

$$p^{\mathrm{req}}_{u,v,q} = p^{\mathrm{req}}_{u,v} f^b_{u,q}. \quad (5)$$

To prevent overload from degrading quality, we cap recommendations at ρ. Thus,

$$\sum\nolimits_{v\in\mathbf{V}} r_{u,v} \le \rho, \quad \forall u \in \mathbf{U}. \quad (6)$$

C. Caching and Request Delivery Model

Let $x^s_{v,q} \in \{0,1\}$ be a binary variable that indicates whether SBS s caches (v,q). where $x^s_{v,q}=1$ means that s caches (v,q). Due to the limited cache capacity of SBS s, the total storage space occupied by the cached video is

$$k_s = \sum\nolimits_{v\in\mathbf{V}}\sum\nolimits_{q\in\mathbf{Q}} x^s_{v,q} b_q L_v, \forall s \in \mathbf{S}, \quad (7)$$

which must not exceed its inherent cache capacity c_s. Moreover, to avoid redundant caching and enhance diversity, we cache at most one bitrate version per video on SBS s, i.e.,

$$\sum\nolimits_{q\in\mathbf{Q}} x^s_{v,q} \le 1. \quad (8)$$

To reflect the multi-level delivery characteristics for MU u requesting (v,q), we define a quintuple $y_{u,v,q} = (y^{\mathrm{d}}_{u,v,q}, y^{\mathrm{t}}_{u,v,q}, y^{\mathrm{n}}_{u,v,q}, y^{\mathrm{s}}_{u,v,q}, y^{\mathrm{c}}_{u,v,q})$. This quintuple consists of five binary variables, each representing a different delivery strategy: direct, transcoding, neighbor, soft, and cloud hits. As MU u's request can be satisfied by only one strategy, we have

$$y^{\mathrm{d}}_{u,v,q} + y^{\mathrm{t}}_{u,v,q} + y^{\mathrm{n}}_{u,v,q} + y^{\mathrm{s}}_{u,v,q} + y^{\mathrm{c}}_{u,v,q} = 1. \quad (9)$$

1) Direct Hit Scenario: When a request of (v,q) arrives, if SBS s_u has the corresponding version, the request is served locally, $y^{\mathrm{d}}_{u,v,q}=1$. The delivery delay comprises only the MU-to-SBS transmission delay; however, since it is ubiquitous across all hit cases [7] and outside our optimization focus, we set the delay $t^{\mathrm{d}}_{u,v,q}=0$.

2) Transcoding Hit Scenario: If the target is not cached in s_u (i.e., $y^{\mathrm{d}}_{u,v,q}=0$), but a higher bitrate version (v,q^+) is available, s_u can transcode (v,q^+) to satisfy the request, $y^{\mathrm{t}}_{u,v,q}=1$. We allow transcoding only from higher to lower bitrate [8]. Then the transcoding delay at s_u is given by

$$t^{\mathrm{trc}}_{u,v,q} = \frac{\sum_{j=1}^{q-1} x^{s_u}_{v,j}\cdot\lambda\cdot(b_j-b_q)\cdot L_v}{f_{s_u}}. \quad (10)$$

Moreover, the total transcoding consumption is expressed as

$$d_s = \sum_{u\in\mathbf{B}_s}\sum_{v\in\mathbf{V}}\sum_{h=2}^{Q} y^{\mathrm{t}}_{u,v,h}\sum_{j=1}^{h-1} x^s_{v,j}\lambda\,(b_j-b_q)\,L_v. \quad (11)$$

Transcoding incurs computational costs. With σ^{trc} as the cost per bit [26], the transcoding cost for SBS s is

$$C^{\mathrm{trc}}_s = \sigma^{\mathrm{trc}} d_s. \quad (12)$$

3) Neighbor Hit Scenario: When s_u does not cache (v,q), it queries its neighboring SBSs. If $\exists s \in \mathbf{N}_{s_u}$ caches (v,q), s_u retrieves the video from this neighbor and forwards it to MU u, i.e., $y^n_{u,v,q}=1$. Let $\hat{s}_u(j^*)$ denote the neighbor with the highest transmission rate among those caching (v,q). The delay for a neighbor hit is then given by $t^{\mathrm{trm}}_{u,v,q} = \frac{b_q L_v}{v_{\hat{s}_u(j^*),s_u}}$, where $v_{\hat{s}_u(j^*),s_u}$ is the transmission rate between $\hat{s}_u(j^*)$ and s_u. The system captures a sequential search prioritizing neighbors with higher transmission rates, and selects the first neighbor that caches the target video [11].

4) Soft Hit Scenario: A soft hit recommends either a lower bitrate version (v,q^-), or a similar version $(\overline{v},q)$, but only if accepted by the MU. For a similar video substitution, the acceptance probability [10] depending on the content similarity $sim_{v,\overline{v}}$ and MU u's preference $p^{\mathrm{pref}}_{u,\overline{v}}$ is $\breve{g}^u_{v,\overline{v}} = p^{\mathrm{pref}}_{u,\overline{v}} sim_{v,\overline{v}}$, and the corresponding soft hit cost is $C^{\mathrm{soft},1}_{u,v,q} = 1 - sim_{v,\overline{v}}$ [19]. For a lower bitrate substitution, the acceptance probability is determined by the MU u's relative bitrate preference as $\hat{g}^u_{q,q^-} = f^b_{u,q^-}/f^b_{u,q}$. Following the QoE model [27], the cost is $C^{\mathrm{soft},2}_{u,v,q} = \log_\beta \frac{q}{q^-}$, where β is a video-specific constant.

To limit MU preference distortion, we introduce threshold a^{soft}_u. Soft hit requires $\breve{g}^u_{v,\overline{v}} \ge a^{\mathrm{soft}}_u$ or $\hat{g}^u_{q,q^-} \ge a^{\mathrm{soft}}_u$. To indicate necessity, we define indicator variable

$$b^{s_u}_{v,q} = \left(1 - x^{s_u}_{v,q}\right)\left[\left(1 - \prod_{i=q+1}^{Q}\left(1 - \mathbb{1}(\breve{g}^u_{v,\overline{v}} \ge a^{\mathrm{soft}}_u)\cdot x^{s_u}_{v,i}\right)\right) + \left(1 - \prod_{j\in V\setminus v}\left(1 - \mathbb{1}(\hat{g}^u_{q,q^-} \ge a^{\mathrm{soft}}_u)\cdot x^{s_u}_{j,q}\right)\right)\right], \quad (13)$$

where $\mathbb{1}(\cdot)$ is the Indicator function. We characterize MU preferences using weights I^c_u and I^r_u, where $I^c_u + I^r_u = 1$. Soft hit selects lower bitrate if $I^c_u > I^r_u$, similar video if $I^c_u < I^r_u$, and random if equal. Expected soft hit cost is Eq. (14).

5) Cloud Hit Scenario: Additionally, the video can be retrieved from the VP's cloud server via the backbone network. The delay is $t^{\mathrm{backhaul}}_{u,v,q} = \frac{b_q L_v}{v_{M,s_u}}$, where v_{M,s_u} denotes the transmission rate between the cloud server and the SBS s_u.

D. Profit Model

The MNO's profit is derived as the difference between the VP's cache leasing revenue and its cache maintenance costs:

$$R^{\mathrm{MNO}}(\delta,\mathbf{x}) = \sum_{s\in\mathbf{S}}(\delta_s - \delta^{\mathrm{op}}_s)\sum_{v\in\mathbf{V}}\sum_{q\in\mathbf{Q}} x^s_{v,q} b_q L_v, \quad (15)$$

where δ^{op}_s denotes the maintenance cost for s.

We adopt a tiered pricing mechanism where the base price for (v,q) is $\sigma_{v,q} = \sigma + (Q-q)\,\sigma_0$, with σ base willingness-to-pay and σ_0 bitrate price gradient. To consider

$$p^{\mathrm{soft}}_{u,v,q} = \sum_{q<j\le Q,\,\mathbb{1}(\breve{g}^u_{v,\overline{v}}\ge a^{\mathrm{soft}}_u)} x^{s_u}_{v,j}\hat{g}^u_{q,j} C^{\mathrm{soft},2}_{u,v,q}\,\mathbb{1}\left(I^c_u \ge I^r_u, \hat{g}^u_{q,q^-} \ge a^{\mathrm{soft}}_u\right) + \sum_{i\in\mathbf{V}\setminus v,\,\mathbb{1}(\hat{g}^u_{v,\overline{v}}\ge a^{\mathrm{soft}}_u)} x^{s_u}_{i,q}\breve{g}^u_{v,i} C^{\mathrm{soft},1}_{u,v,q}\,\mathbb{1}\left(I^c_u < I^r_u, \breve{g}^u_{v,\overline{v}} \ge a^{\mathrm{soft}}_u\right) \quad (14)$$

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

service delay, we apply a reward-penalty method [28] based on the delivery deadline $t^{\mathrm{ddl}}_{u,v,q}$. Define the time difference as $\Delta t_{u,v,q} = t^{\mathrm{ddl}}_{u,v,q} - t_{u,v,q}$,yielding the final MU payment $\sigma_{v,q}\left(1+\epsilon\cdot\frac{\Delta t_{u,v,q}}{t^{\mathrm{ddl}}_{u,v,q}}\right)$, where ϵ is the reward-penalty weight. The VP revenues of different hit scenarios are as follows.

① Direct Hit Scenario: With $t_{u,v,q}=0$, the revenue is

$$R^{\mathrm{VP,d}}_{u,v,q} = y^{\mathrm{d}}_{u,v,q} x^{s_u}_{v,q} \sigma_{v,q}(1+\epsilon). \tag{16}$$

② Transcoding Hit Scenario: With $t_{u,v,q} = t^{\mathrm{trc}}_{u,v,q}$, the revenue is

$$R^{\mathrm{VP,t}}_{u,v,q} = y^{\mathrm{t}}_{u,v,q}\cdot\sigma_{v,q}\left(1+\epsilon\cdot\frac{t^{\mathrm{ddl}}_{u,v,q}-t^{\mathrm{trc}}_{u,v,q}}{t^{\mathrm{ddl}}_{u,v,q}}\right). \tag{17}$$

③ Neighbor Hit Scenario: With $t_{u,v,q} = t^{\mathrm{trm}}_{u,v,q}$, the revenue is

$$R^{\mathrm{VP,n}}_{u,v,q} = y^{\mathrm{n}}_{u,v,q}\cdot\sigma_{v,q}\left(1+\epsilon\cdot\frac{t^{\mathrm{ddl}}_{u,v,q}-t^{\mathrm{trm}}_{u,v,q}}{t^{\mathrm{ddl}}_{u,v,q}}\right). \tag{18}$$

④ Soft Hit Scenario: With $t_{u,v,q}=0$, but incurring similarity cost, the expected MU payment is Eq. (19). The corresponding revenue is $R^{\mathrm{VP,s}}_{u,v,q} = y^{\mathrm{s}}_{u,v,q} R^{\mathrm{soft}}_{u,v,q}$.

⑤ Cloud Hit Scenario: For cloud hit, the revenue is

$$R^{\mathrm{VP,c}}_{u,v,q} = y^{\mathrm{c}}_{u,v,q}\cdot\sigma_{v,q}\left(1+\epsilon\cdot\frac{t^{\mathrm{ddl}}_{u,v,q}-t^{\mathrm{backhaul}}_{u,v,q}}{t^{\mathrm{ddl}}_{u,v,q}}\right). \tag{20}$$

Aggregating all scenarios, the VP revenue is $R^{\mathrm{VP}}_{u,v,q} = R^{\mathrm{VP,d}}_{u,v,q} + R^{\mathrm{VP,t}}_{u,v,q} + R^{\mathrm{VP,n}}_{u,v,q} + R^{\mathrm{VP,s}}_{u,v,q} + R^{\mathrm{VP,c}}_{u,v,q}$. Then, the total VP profit, calculated as MU payments minus similarity cost, leasing fees, and transcoding costs, is expressed as

$$\begin{aligned} R^{\mathrm{VP}}(\delta,\mathbf{x},\mathbf{r},\mathbf{y}) = &\sum_{u\in\mathbf{U}}\sum_{v\in\mathbf{V}}\sum_{q\in\mathbf{Q}} p^{\mathrm{req}}_{u,v,q} R^{\mathrm{VP}}_{u,v,q} - y^{\mathrm{s}}_{u,v,q}\eta p^{\mathrm{soft}}_{u,v,q} \\ &- \sum_{s\in\mathbf{S}}\delta_s \sum_{v\in\mathbf{V}}\sum_{q\in\mathbf{Q}} x^s_{v,q} b_q L_v - \sum_{s\in\mathbf{S}} C^{\mathrm{trc}}_s, \end{aligned} \tag{21}$$

where η is the weight parameter for similarity cost, used to align the units of MU payment and similarity cost.

E. Problem Formulation

This section formulates the profit-maximization problem for an MNO and a VP to collaboratively orchestrate video recommendation, edge cache, and content delivery. The MNO sets a cache-leasing price δ per unit capacity; the VP then adapts its policies. This Stackelberg game has MNO as leader and VP as follower. The MNO's objective is to choose an optimal δ to maximize its profit, calculated as the cache-leasing income minus maintenance costs. Formally, the leader problem is

$$\textbf{P1:}\quad \max_{\delta}\ R^{\mathrm{MNO}}(\delta,\mathbf{x}) \tag{22}$$

$$\text{s.t.}\quad 0\le\delta\le\delta_{\max}, \tag{22.a}$$

where Eq. (22.a) restricts the MNO's cache pricing and $\delta_{\max}$ is the upper bound.

Given the MNO's δ, the VP jointly optimizes its decisions to maximize its profit. The VP's problem is defined as

$$\textbf{P2:}\quad \max_{\mathbf{x},\mathbf{r},\mathbf{y}}\ R^{\mathrm{VP}}(\delta,\mathbf{x},\mathbf{r},\mathbf{y}), \tag{23}$$

$$\text{s.t.}\quad r_{u,v} \le \mathbb{1}(p^{\mathrm{pref}}_{u,v} \ge a^{\mathrm{rec}}_u), \tag{23.a}$$

$$y^d_{u,v,q} \le \mathbb{1}(x^{s_u}_{v,q}=1), \tag{23.b}$$

$$y^{\mathrm{t}}_{u,v,q} \le \left(1-x^{s_u}_{v,q}\right)\left(1-\prod_{j=1}^{q-1}\left(1-x^{s_u}_{v,j}\right)\right), \tag{23.c}$$

$$y^{\mathrm{n}}_{u,v,q} \le \left(1-x^{s_u}_{v,q}\right)\left(1-\prod_{s\in N_{s_u}}\left(1-x^{s}_{v,q}\right)\right), \tag{23.d}$$

$$y^{\mathrm{s}}_{u,v,q} \le b^{s_u}_{v,q}, \tag{23.e}$$

$$0\le k_s\le c_s,\ 0\le d_s\le D_s, \tag{23.f}$$

$$y^{\mathrm{d}}_{u,v,q}, y^{\mathrm{t}}_{u,v,q}, y^{\mathrm{n}}_{u,v,q}, y^{\mathrm{s}}_{u,v,q}, y^{\mathrm{c}}_{u,v,q} \in \{0,1\}, \tag{23.g}$$

(6)(8)(9)

where Eq. (23.a) stipulates that recommendation requires video's preference surpasses the recommendation threshold a^{rec}_u. Eq. (23.b) enables direct hit, i.e., $y^{\mathrm{d}}_{u,v,q}=1$ when SBS s_u caches (v,q). Eq. (23.c) enables transcoding hit, i.e., $y^{\mathrm{t}}_{u,v,q}=1$ when SBS s_u does not cache (v,q) but possesses a higher bitrate version (v,q^+). Eq. (23.d) enables neighbor hit, i.e., $y^{\mathrm{n}}_{u,v,q}=1$ iff SBS s_u does not cache (v,q) while at least one neighbor caches it. Eq. (23.e) enables soft hit when $b^{s_u}_{v,q}\in\{1,2\}$. Eq. (23.f) is the cache capacity and computation limits. Eq. (23.g) lists all binary variables.

III. Algorithm Description

A. Methodology Overview

We propose a backward induction-based optimization scheme (BIS) to jointly optimize the profits of MNO and VP by achieving the Stackelberg Equilibrium. BIS first addresses VP's problem **P2** for the optimal response strategies involving video recommendation $\mathbf{r}^*(\delta)$, cache placement $\mathbf{x}^*(\delta)$, and request delivery $\mathbf{y}^*(\delta)$ under a given basic cache-leasing price δ. Then, BIS solves the MNO's problem **P1** for the optimal δ^* by utilizing the acquired $\mathbf{r}^*(\delta)$, $\mathbf{x}^*(\delta)$, and $\mathbf{y}^*(\delta)$. **P2** is a non-convex integer program with three interdependent binary decision sets. BIS tackles it via a block coordinate descent (BCD), dividing **P2** into three coupled subproblems **P3** (recommendation), **P4** (cache placement), and **P5** (delivery). To solve these subproblems, we propose a sequential layered video recommendation algorithm, a relaxation and rounding-based cache placement algorithm, and a relaxation and rounding-based request delivery algorithm. The BCD iteratively uses these algorithms to optimize VP's profit. For the MNO, the profit R^{MNO} initially rises and subsequently declines as δ increases, as observed in [11]. Thus, we propose a peak-seeking pricing strategy to optimize MNO's profit. The details are provided below.

B. Sequential Layered Video Recommendation

This section addresses the video recommendation subproblem to optimize VP's profit. Given the basic δ, the cache

$$R^{\mathrm{soft}}_{u,v,q} = \sum_{q<j\le Q} x^{s_u}_{v,j}\hat{g}^u_{q,j}\left(\sigma_{v,j}(1+\epsilon)\right)\mathbb{1}\left(I^c_u\ge I^r_u, \hat{g}^u_{q,q^-}\ge a^{\mathrm{soft}}_u\right) + \sum_{i\in\mathbf{V}\setminus v} x^{s_u}_{i,q}\check{g}^u_{v,i}\left(\sigma_{v,q}(1+\epsilon)\right)\mathbb{1}\left(I^c_u< I^r_u, \check{g}^u_{v,\overline{v}}\ge a^{\mathrm{soft}}_u\right) \tag{19}$$

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

placement $\mathbf{x}$, and the request delivery $\mathbf{y}$, the VP's profit R^{VP} with respect to $\mathbf{r}$ is

$$R_1^{\mathrm{VP}}(\mathbf{r}) = \sum\nolimits_{u\in\mathbf{U}}\sum\nolimits_{v\in\mathbf{V}}\left(\alpha_{u,v}^{1,1}\frac{r_{u,v}}{\sum_{i\in\mathbf{V}} r_{u,i}p_{u,i}^{\mathrm{pref}}} + \alpha_{u,v}^{1,2}\frac{1-r_{u,v}}{\sum_{i\in V}(1-r_{u,i})p_{u,i}^{\mathrm{pref}}}\right) + \beta_0, \tag{24}$$

where $\alpha_{u,v}^{1,1}$, $\alpha_{u,v}^{1,2}$, and β_0 are constants:

$$\begin{aligned}
\alpha_{u,v}^{1,1} =& o_u p_{u,v}^{\mathrm{pref}} \sum\nolimits_{q\in\mathbf{Q}} f_{u,q}^{b} R_{u,v,q}^{\mathrm{VP}},\\
\alpha_{u,v}^{1,2} =& (1-o_u) p_{u,v}^{\mathrm{pref}} \sum\nolimits_{q\in\mathbf{Q}} f_{u,q}^{b} R_{u,v,q}^{\mathrm{VP}},\\
\beta_0 =& -\sum\nolimits_{s\in\mathbf{S}}\delta_s\sum\nolimits_{v\in\mathbf{V}}\sum\nolimits_{q\in\mathbf{Q}} x_v^{s,q}p_{v,q}\\
&-\sum\nolimits_{s\in\mathbf{S}}\sigma^{\mathrm{trc}}\sum\nolimits_{u\in\mathbf{B}_s}\sum\nolimits_{v\in\mathbf{V}}\sum\nolimits_{h=2}^{Q} y_{u,v,h}^{\mathrm{t}}\\
&\times\sum\nolimits_{j=1}^{h-1} x_{v,h}^{s}\lambda(b_j-b_h)L_v.
\end{aligned}$$

The video recommendation problem is defined as

$$\textbf{P3:}\quad \max_{\mathbf{r}} R_1^{\mathrm{VP}}(\mathbf{r})$$
$$\text{s.t.}\quad (6), (23.a), (23.g).$$

As shown in Eq. (24), $R_1^{\mathrm{VP}}(\mathbf{r})$ is nonlinear in $\mathbf{r}$, making **P3** difficult to solve directly. We propose a sequential layered video recommendation algorithm leveraging MU preferences and cache placement. For each MU u, the algorithm first establishes a recommended video candidate set V_u^{rec} where preference $p_{u,v}^{\mathrm{pref}}$ is larger than a_u^{rec} to ensure that the recommended videos are consistent with the MU u's preferences. The videos in V_u^{rec} are sorted in descending order of $p_{u,v}^{\mathrm{pref}}$ so that higher preferences videos are recommended first. Considering that different SBSs cache various videos, we follow a sequential layered recommendation: first recommend videos cached in the associated SBS s_u, then videos cached in neighboring SBSs of s_u, and finally other videos in V_u^{rec}.

The pseudo-code of the sequential layered video recommendation algorithm is shown in **Algorithm 1**. The algorithm first initializes all variables in $\mathbf{r}$ to 0 (line 1). For each $u \in U$, line 3 establishes a candidate set V_u^{rec}, and line 4 sorts videos descending by $p_{u,v}^{\mathrm{pref}}$. Lines 5-7 operate the first iteration through V_u^{rec}. If a video v is cached by SBS s_u and recommendations are insufficient (i.e., $\sum_v r_{u,v} < \rho$), we update the video recommendation $r_{u,v}$ and V_u^{rec} to avoid repetition. If $\sum_v r_{u,v} < \rho$ still holds, we proceed to a second iteration through V_u^{rec} (lines 8-10). The recommendation $r_{u,v}$ and the candidate set V_u^{rec} are updated if a video v is cached by an SBS neighboring s_u and recommendations remain inadequate. After the second iteration, if $\sum_v r_{u,v} < \rho$ still holds, the algorithm goes through V_u^{rec} a third time, continuing to recommend videos until $\sum_v r_{u,v} = \rho$ (lines 11-12).

Algorithm 1: Sequential Layered Video Recommendation

Input : MU set $\mathbf{U}$, video set $\mathbf{V}$, recommendation number limit ρ, basic cache-leasing price δ, cache placement $\mathbf{x}$, and request delivery $\mathbf{y}$
Output: Video recommendation $\mathbf{r}$

1 Initialize each variables in $\mathbf{r}$ to 0;
2 **for** each $u \in \mathbf{U}$ **do**
3 Set $V_u^{\mathrm{rec}} = \{v \mid p_{u,v}^{\mathrm{pref}} \geq a_u^{\mathrm{rec}}\}$;
4 Sort videos in V_u^{rec} in descending order of $p_{u,v}^{\mathrm{pref}}$;
5 **for** each $v \in V_u^{\mathrm{rec}}$ **do**
6 **if** $\exists q \in \mathbf{Q}$ such that $x_{v,q}^{s_u} = 1$ and $\sum_v r_{u,v} < \rho$ **then**
7 Set $r_{u,v} = 1$ and $V_u^{\mathrm{rec}} = V_u^{\mathrm{rec}} \setminus \{v\}$;
8 **for** each $v \in V_u^{\mathrm{rec}}$ **do**
9 **if** $\exists q \in \mathbf{Q}$ and $\exists s \in \mathbf{N}_{s_u}$ such that $x_{v,q}^{s} = 1$ and $\sum_v r_{u,v} < \rho$ **then**
10 Set $r_{u,v} = 1$ and $V_u^{\mathrm{rec}} = V_u^{\mathrm{rec}} \setminus \{v\}$;
11 **while** $\sum_v r_{u,v} < \rho$ **do**
12 Set $r_{u,v} = 1$ and $V_u^{\mathrm{rec}} = V_u^{\mathrm{rec}} \setminus \{v\}$;

C. Relaxation and Rounding-based Cache placement

This section addresses the cache placement subproblem to maximize VP's profit while ensuring MU QoE by rational allocation of limited SBS cache resources. We introduce an auxiliary variable $\pi_{u,v,q,j} = x_{v,q}^{\hat{s}_u(j)}\prod_{k=0}^{j-1}(1-x_{v,q}^{\hat{s}_u(k)})$, which means the j-th neighbor is the first to cache (v,q). Given the basic cache-leasing price δ, the video recommendation $\mathbf{r}$, and the request delivery $\mathbf{y}$, the VP's profit R^{VP} with respect to the cache placement $\mathbf{x}$ and the auxiliary variable $\pi_{u,v,q,j}$ is

$$\begin{aligned}
R_2^{\mathrm{VP}}(\mathbf{x},\boldsymbol{\pi}) =& \sum\nolimits_{u\in\mathbf{U}}\sum\nolimits_{v\in\mathbf{V}}\sum\nolimits_{q\in Q}[\alpha^{2,1}x_{v,q}^{s_u}\\
&-\alpha^{2,2}\sum\nolimits_{j=1}^{q-1}\alpha_j^{2,7}x_{v,j}^{s_u} - \sum\nolimits_{j\in\hat{s}_u}\alpha_j^{2,3}\pi_{u,v,q,j}\\
&+\sum\nolimits_{q<j\leq Q}\alpha_j^{2,4}x_{v,j}^{s_u} + \sum\nolimits_{i\in\mathbf{V}}\alpha_i^{2,5}x_{i,q}^{s_u} + \beta_1]\\
&-\sum\nolimits_{s\in\mathbf{S}}\sum\nolimits_{v\in\mathbf{V}}\sum\nolimits_{q\in Q}\alpha_s^{2,6}x_{v,q}^{s}\\
&-\sum_{s\in\mathbf{S}}\sum_{u\in\mathbf{B}_s}\sum_{k\in\mathbf{V}}\sum\nolimits_{q=2}^{Q} y_{u,k,h}^{t}\sum\nolimits_{j=1}^{q-1}\sigma^{\mathrm{trc}}\alpha_j^{2,7}x_{k,j}^{s},
\end{aligned}\tag{25}$$

where $\alpha_{u,v,q}^{2,1}$, $\alpha_{u,v,q}^{2,2}$, $\alpha_{u,v,q,j}^{2,3}$, $\alpha_{u,v,q,j}^{2,4}$, $\alpha_{u,v,q,j}^{2,5}$, $\alpha_{s,v,q}^{2,6}$, $\alpha_{j,h,k}^{2,7}$, and β_1 are constants:

$$\begin{aligned}
\alpha_{u,v,q}^{2,1} =& p_{u,v,q}^{\mathrm{req}} y_{u,v,q}^{d}\sigma_{v,q}(1+\epsilon),\\
\alpha_{u,v,q}^{2,2} =& p_{u,v,q}^{\mathrm{req}} y_{u,v,q}^{t}\sigma_{v,q}\epsilon/(t_{u,v,q}^{\mathrm{ddl}} f_{s_u}),\\
\alpha_{u,v,q,j}^{2,3} =& p_{u,v,q}^{\mathrm{req}} y_{u,v,q}^{n}\sigma_{v,q}\epsilon d_{\hat{s}_u(j),s_u}/t_{u,v,q}^{\mathrm{ddl}},\\
\alpha_{u,v,q,j}^{2,4} =& y_{u,v,q}^{\mathrm{s}}\hat{g}_{q,j}^{u}\left(\sigma_{v,j}(1+\epsilon) - \eta\log_\beta\frac{q}{j}\right)\\
&\times\left(\mathbb{1}(I_u^c - I_u^r) + kr_{I_u^c I_u^r}\mathbb{1}(\mathrm{rand}-0.5)\right),\\
\alpha_{u,v,q,j}^{2,5} =& y_{u,v,q}^{\mathrm{s}}\hat{g}_{v,i}^{u}\left(\sigma_{v,j}(1+\epsilon) - \eta(1-\mathrm{sim}(v,i))\right)\\
&\times\left(\mathbb{1}(I_u^c - I_u^r) + kr_{I_u^c I_u^r}\mathbb{1}(\mathrm{rand}-0.5)\right),\\
\alpha_{s,v,q}^{2,6} =& \delta_s b_q L_v, \alpha_{j,h,k}^{2,7} = \lambda(b_j - b_h)L_k,\\
\beta_1 =& p_{u,v,q}^{\mathrm{req}} y_{u,v,q}^{\mathrm{t}}\sigma_{v,q}(1+\epsilon) + p_{u,v,q}^{\mathrm{req}} y_{u,v,q}^{\mathrm{n}}\sigma_{v,q}(1+\epsilon)\\
&+ y_{u,v,q}^{\mathrm{c}}\sigma_{v,q}\left(1+\epsilon\frac{t_{u,v,q}^{\mathrm{ddl}} - t_{u,v,q}^{\mathrm{backhaul}}}{t_{u,v,q}^{\mathrm{ddl}}}\right).
\end{aligned}$$

The cache placement problem is defined as

$$\textbf{P4:}\quad \max_{\mathbf{x},\boldsymbol{\pi}} R_2^{\mathrm{VP}}(\mathbf{x},\boldsymbol{\pi}), \tag{26}$$
$$\text{s.t.}\quad \pi_{u,v,q,j} \leq 1 - x_{v,q}^{\hat{s}_u(k)}, \forall k < j, 1 \leq j \leq N, \tag{26.a}$$
$$\pi_{u,v,q,j} \geq x_{v,q}^{\hat{s}_u(j)} - \sum\nolimits_{k=0}^{j-1}\left(1 - x_{v,q}^{\hat{s}_u(k)}\right), \tag{26.b}$$

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: Relaxation and Rounding-based Caching Placement

Input : MU set $\mathbf{U}$, video set $\mathbf{V}$, SBS set $\mathbf{S}$, bitrate set $\mathbf{Q}$, basic cache-leasing price δ, video recommendation $\mathbf{r}$, and request delivery $\mathbf{y}$

Output: Caching placement $\mathbf{x}$

1 Obtain the optimal relaxed solution $(\hat{\mathbf{x}}, \hat{\Pi})$ by a solver;
2 **for** each $s \in \mathbf{S}$ **do**
3 Initialize caching placement set $\mathbb{C}_s = \emptyset$;
4 Initialize used cache capacity $c_s^{\text{used}} = 0$;
5 Get video version's unit storage value $\varsigma_{v,q}^s$ by Eq. (27);
6 **for** each $v \in \mathbf{V}$ **do**
7 Find $q^* = \arg\max_{q \in \{q \in \mathbf{Q} | \varsigma_{v,q}^s > 0\}} \hat{x}_{v,q}^s$;
8 Update $\mathbb{C}_s = \mathbb{C}_s \cup \{(v, q^*)\}$, $c_s^{\text{used}} = c_s^{\text{used}} + L_v b_{q^*}$, and $x_{v,q^*}^s = 1$;
9 Sort videos in $\mathbb{C}_s$ in the ascending order of $\varsigma_{v,q}^s$;
10 **while** $c_s^{used} > c_s$ **do**
11 Obtain the first video version (v, q) in $\mathbb{C}_s$;
12 Update $\mathbb{C}_s = \mathbb{C}_s \backslash \{(v, q)\}$, $c_s^{\text{used}} = c_s^{\text{used}} - L_v b_q$, and $x_{v,q}^s = 0$;

$$\sum\nolimits_{j \in \hat{s}_u} \pi_{u,v,q,j} \le 1, \tag{26.c}$$

$$\pi_{u,v,q,j}, x_{v,q}^s \in \{0, 1\}, \tag{26.d}$$

(8)(23.f)

where Eq. (26.a) ensures that the auxiliary variable $\pi_{u,v,q,j}$ takes 1 iff the j-th neighbor actually caches the requested (v, q) and no earlier neighbor does. Eq. (26.b) prevents activation of preceding higher-rate neighbors before j. Eq. (26.c) limits each request to at most one neighbor hit. Eq. (26.d) is the binary constraint for the cache placement decisions and auxiliary variables.

P4 is a non-convex integer programming problem, which is hard to solve directly. We propose a relaxation and rounding-based cache placement algorithm. It relaxes binary variables $x_{v,q}^s$ and $\pi_{u,v,q,j}$ to continuous in the interval $[0, 1]$ and utilizes the standard optimization solvers (e.g., Gurobi) to obtain the optimal relaxed solution $(\hat{\mathbf{x}}, \hat{\Pi})$. For SBS s, we calculate the unit storage value of each version (v, q):

$$\varsigma_{v,q}^s = \frac{\partial R_2^{\text{VP}}}{\partial x_{v,q}^s} / L_v b_q. \tag{27}$$

For each video v, SBS s caches the version with the maximum relaxed solution and positive unit storage value. If the total used capacity exceeds SBS s's capacity, we iteratively remove the video with the lowest unit storage value.

Algorithm 2 gives the pseudo-code of the relaxation and rounding-based cache placement algorithm. It first obtains the optimal relaxed solution $(\hat{\mathbf{x}}, \hat{\Pi})$ using a standard optimization solver (line 1). Lines 2-12 deploy video versions for each SBS to optimize R_2^{VP}. Specifically, for SBS s, the caching set $\mathbb{C}_s$ is initialized empty and the used capacity $c_s^{\text{used}} = 0$ in lines 3-4. Line 5 calculates the unit storage value of each video version using Eq. (27). For each video v, line 7 finds version (v, q^*) with the maximum relaxed solution and a positive value. Then, the algorithm caches the version (v, q^*) at SBS s, and updates the caching set $\mathbb{C}_s$, the used caching capacity c_s^{used}, and the caching placement metric x_{v,q^*}^s (line 8). Finally, if the used capacity exceeds SBS s's capacity, the video version with the lowest unit storage value is eliminated gradually (lines 9-12).

D. Relaxation and Rounding-based Request Delivery

This section focuses on the request delivery subproblem, aiming to maximize VP's profit while satisfying system resource constraints. Given the basic cache-leasing price δ, the video recommendation $\mathbf{r}$, and the cache placement $\mathbf{x}$, the VP's profit function R^{VP} with respect to the request delivery $\mathbf{y}$ is

$$\begin{aligned} R_3^{\text{VP}}(\mathbf{y}) = &\sum\nolimits_{u \in \mathbf{U}} \sum\nolimits_{v \in \mathbf{V}} \sum\nolimits_{q \in \mathbf{Q}} \Big(\alpha_{u,v,q}^{3,1} y_{u,v,q}^{\text{d}} \\ &- \alpha_{u,v,q}^{3,2} y_{u,v,q}^{\text{t}} - \alpha_{u,v,q}^{3,3} y_{u,v,q}^{\text{n}} + \beta_2 (y_{u,v,q}^{\text{t}} + y_{u,v,q}^{\text{n}}) \\ &+ \alpha_{u,v,q}^{3,4} y_{u,v,q}^{\text{s}} + \alpha_{u,v,q}^{3,5} y_{u,v,q}^{\text{c}} \Big) - \sum\nolimits_{s \in \mathbf{S}} \sigma^{\text{trc}} \\ &\times \sum\nolimits_{u \in \mathbf{B}_s} \sum\nolimits_{k \in \mathbf{V}} \sum\nolimits_{h=2}^{Q} y_{u,k,h}^{\text{t}} \alpha_{k,h}^{3,6} + \beta_3, \end{aligned} \tag{28}$$

where $\alpha_{u,v,q}^{3,1}$, $\alpha_{u,v,q}^{3,2}$, $\alpha_{u,v,q}^{3,3}$, $\alpha_{u,v,q}^{3,4}$, $\alpha_{u,v,q}^{3,5}$, $\alpha_{k,h}^{3,6}$, β_2, and β_3 are constants:

$$\begin{aligned} \alpha_{u,v,q}^{3,1} =& p_{u,v,q}^{\text{req}} x_{v,q}^{s_u} \sigma_{v,q} (1 + \epsilon), \\ \alpha_{u,v,q}^{3,2} =& p_{u,v,q}^{\text{req}} \sigma_{v,q} \epsilon \sum\nolimits_{j=1}^{h-1} x_{k,j}^{s_u} \lambda (b_j - b_h) L_k \big/ (t_{u,v,q}^{\text{ddl}} f_{s_u}), \\ \alpha_{u,v,q}^{3,3} =& p_{u,v,q}^{\text{req}} \sigma_{v,q} \epsilon \sum\nolimits_{j \in \hat{s}_u} \left[\prod\nolimits_{k=0}^{j-1} \left(1 - x_{v,q}^{\hat{s}_u(k)} \right) \right] \\ &\times x_{v,q}^{\hat{s}_u(j)} d_{\hat{s}_u(j), s_u} \Big/ t_{u,v,q}^{\text{ddl}}, \\ \alpha_{u,v,q}^{3,4} =& \sum\nolimits_{q < j \le Q} x_{v,j}^{s_u} \hat{g}_{q,j}^u \left(\sigma_{v,j} (1 + \epsilon) - \eta \log_\beta \frac{q}{j} \right) \\ &\times \left(\mathbb{1}(I_u^c - I_u^r) + k r_{I_u^r, I_u^r} \mathbb{1}(\text{rand} - 0.5) \right) \\ &+ \sum\nolimits_{i \in \mathbf{V}} x_{l,q}^{s_u} \cdot \check{g}_{v,i}^u \left(\sigma_{v,j} (1 + \epsilon) - \eta (1 - \text{sim}(v, i)) \right) \\ &\times \left(\mathbb{1}(I_u^r - I_u^c) + k r_{I_u^r, I_u^r} (1 - \mathbb{1}(\text{rand} - 0.5)) \right), \\ \alpha_{u,v,q}^{3,5} =& \sigma_{v,q} \left(1 + \epsilon \frac{t_{u,v,q}^{\text{ddl}} - t_{u,v,q}^{\text{backhaul}}}{t_{u,v,q}^{\text{ddl}}} \right), \\ \alpha_{k,h}^{3,6} =& \sum\nolimits_{j=1}^{h-1} x_{k,j}^{s_u} \lambda (b_j - b_h) L_k, \\ \beta_2 =& p_{u,v,q}^{\text{req}} \sigma_{v,q} (1 + \epsilon), \\ \beta_3 =& - \sum\nolimits_{s \in \mathbf{S}} \delta_s \sum\nolimits_{v \in \mathbf{V}} \sum\nolimits_{q \in \mathbf{Q}} x_{v,q}^s b_q L_v. \end{aligned}$$

The request delivery problem is defined as

$$\begin{aligned} \textbf{P5:} \quad & \max_{\mathbf{y}} R_3^{\text{VP}}(\mathbf{y}) \\ \text{s.t.} \quad & (9), (23.\text{b})\text{-}(23.\text{f}). \end{aligned} \tag{29}$$

P5 is a multi-variable linear integer programming problem with high computational complexity. We propose a relaxation and rounding-based request delivery algorithm. The algorithm first relaxes the binary variables $y_{u,v,q}^{\text{d}}$, $y_{u,v,q}^{\text{t}}$, $y_{u,v,q}^{\text{n}}$, $y_{u,v,q}^{\text{s}}$, and $y_{u,v,q}^{\text{c}}$ into continuous variables within $[0, 1]$ and uses the standard optimization solvers to solve the relaxed problem. Then, for each MU-video-bitrate combination (u, v, q), we set the maximum element in the relaxed solution $\widehat{y_{u,v,q}}$ to 1 and the rest to 0. If the transcoding hit method is employed ($y_{u,v,q}^{\text{t}} = 1$) and the computation resource is insufficient, we set the second-maximum element to 1 and the rest to 0.

The pseudo-code of the relaxation and rounding-based request delivery algorithm is shown in **Algorithm 3**. The

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3: Relaxation and Rounding-based Request Delivery

Input : MU set $\mathbf{U}$, video set $\mathbf{V}$, SBS set $\mathbf{S}$, bitrate set $\mathbf{Q}$, basic cache-leasing price δ, video recommendation $\mathbf{r}$, and cache placement $\mathbf{x}$
Output: Request delivery $\mathbf{y}$

1 Obtain the optimal relaxed solution $\hat{\mathbf{y}}$ by a solver;
2 Initialize $D^{s}_{\text{used}} = 0$ for each SBS $s \in \mathbf{S}$;
3 Initialize each elements in $\mathbf{y}$ to 0;
4 **for** each $u \in \mathbf{U}$, $v \in \mathbf{V}$, $q \in \mathbf{Q}$ **do**
5 Obtain $idx_1 = \arg\max_{i \in \{1,2,\cdots,5\}} \widehat{y_{u,v,q}}(i)$ and $idx_2 = \arg\max_{i \neq idx_1} \widehat{y_{u,v,q}}(i)$;
6 Set $idx = idx_1$;
7 **if** $idx == 2$ **then**
8 Calculate the computation resource d_{s_u} by Eq. (11);
9 **if** $D^{s_u}_{\text{used}} + d_{s_u} \leq D_{s_u}$ **then**
10 Set $y^{\text{t}}_{u,v,q} = 1$;
11 Update $D^{s_u}_{\text{used}} = D^{s_u}_{\text{used}} + d_{s_u}$;
12 **else**
13 Set $idx = idx_2$;
14 **if** $idx == 1$ **then**
15 Set $y^{\text{d}}_{u,v,q} = 1$;
16 **else if** $idx == 3$ **then**
17 Set $y^{\text{n}}_{u,v,q} = 1$;
18 **else if** $idx == 4$ **then**
19 Set $y^{\text{s}}_{u,v,q} = 1$;
20 **else if** $idx == 5$ **then**
21 Set $y^{\text{c}}_{u,v,q} = 1$;

algorithm first obtains the optimal relaxed solution $\hat{\mathbf{y}}$ for **P5** using a standard optimization solver (line 1). Line 2 initializes the used computation resource D^{s}_{used} to 0 for each SBS $s \in \mathbf{S}$. Each element in the request delivery $\mathbf{y}$ is initialized to 0 in line 3. Lines 4-22 decide the delivery strategy for each MU-video-bitrate combination (u, v, q). Specifically, line 5 finds the indices of the maximum and the second-maximum elements in the relaxed solution $\widehat{y_{u,v,q}}$, denoted as idx_1 and idx_2, respectively. The algorithm selects the index of the element with the maximum value (line 6). If the selected delivery method is the transconding hit method ($idx = 2$), we need to judge whether the computation resource is sufficient. Line 8 calculates the required computation resource d_{s_u} by Eq. (11). If the resource is insufficient, the algorithm updates the used computation resource $D^{s_u}_{\text{used}}$ of the associated SBS s_u and set the delivery element $y^{\text{t}}_{u,v,q}$ to 1 (lines 9-11). Otherwise, the index of the element with the second-maximum value is selected (lines 12-13). Finally, lines 14-21 set the delivery element corresponding to the selected index to 1.

E. BCD-based Optimization for VP

After solving the video recommendation, caching placement, and request delivery subproblems, BIS utilizes a BCD-based optimization algorithm (**Algorithm 4**) to maximize VP's profit under the given basic cache-leasing price δ. Given δ, the algorithm first initializes the maximum number of iterations $\mathcal{I}^{\text{VP}}$ and the convergence threshold φ (line 1). Then, line

Algorithm 4: BCD-based Optimization

Input : MU set $\mathbf{U}$, video set $\mathbf{V}$, SBS set $\mathbf{S}$, bitrate set $\mathbf{Q}$, and basic cache-leasing price δ
Output: Optimal strategies $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ and profit R^{VP}

1 Initialize the maximum iterations number $\mathcal{I}^{\text{VP}}$ and the convergence threshold φ;
2 Generate strategy $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ randomly;
3 **for** $i = 1$ **to** $\mathcal{I}^{\text{VP}}$ **do**
4 Call **Algorithm 1** to update video recommendation $\mathbf{r}$;
5 Call **Algorithm 2** to update cache placement $\mathbf{x}$;
6 Call **Algorithm 3** to update request delivery $\mathbf{y}$;
7 Calculate VP profit $R^{\text{VP}}(i)$ by Eq. (21);
8 **if** $R^{\text{VP}}(i) - R^{\text{VP}}(i-1) < \varphi$ **then**
9 **break**;

Algorithm 5: Peak-seeking Pricing Strategy for MNO

Input : Video set $\mathbf{V}$, MU set $\mathbf{U}$, SBS set $\mathbf{S}$, bitrate set $\mathbf{Q}$, and the maximum basic cache-leasing price $\delta_{\max}$
Output: Optimal basic cache-leasing price δ^* and the corresponding MNO profit R^{MNO^*}

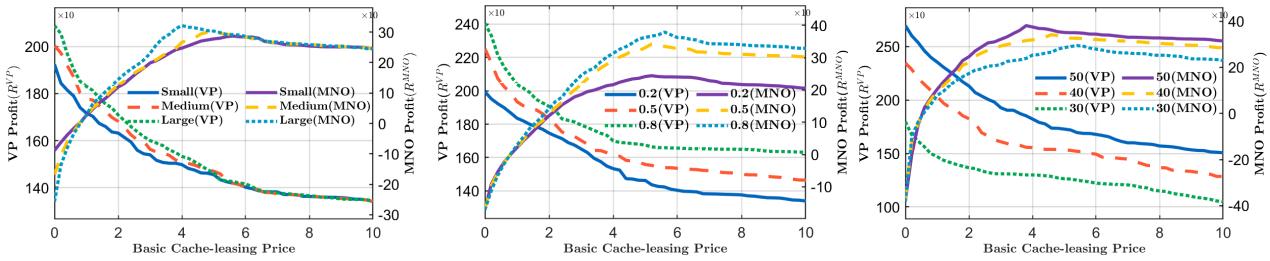
1 Initialize the price adjustment step size $\Delta\delta$;
2 Calculate the maximum iteration number $\mathcal{I}^{\text{MNO}} = \lceil \frac{\delta_{\max}}{\Delta\delta} \rceil$;
3 **for** $i = 1$ **to** $\mathcal{I}^{\text{MNO}}$ **do**
4 Set $\delta = (i-1)\Delta\delta$;
5 Call **Algorithm 4** to obtain VP strategy $(\mathbf{x}, \mathbf{r}, \mathbf{y})$;
6 Calculate MNO profit $R^{\text{MNO}}(i)$ by Eq. (15);
7 **if** $R^{\text{MNO}}(i) \leq R^{\text{MNO}}(i-1)$ **then**
8 **break**;
9 **if** $i < \mathcal{I}^{\text{MNO}}$ **then**
10 Set $\delta^* = (i-2)\Delta\delta$ and $R^{\text{MNO}^*} = R^{\text{MNO}}(i-2)$;
11 **else**
12 Set $\delta^* = (i-1)\Delta\delta$ and $R^{\text{MNO}^*} = R^{\text{MNO}}(i-1)$;

2 generates the initial strategy $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ randomly. Lines 3-9 iteratively update the video recommendation $\mathbf{r}$, cache placement $\mathbf{x}$, and request delivery $\mathbf{y}$ until the maximum number of iterations $\mathcal{I}^{\text{VP}}$ is reached or the change in VP's profit R^{VP} is below the convergence threshold φ. Specifically, in each iteration, lines 4-6 call **Algorithms 1-3** to update the video recommendation, cache placement, and request delivery, respectively. Line 7 derives the VP's profit $R^{\text{VP}}(i)$ under the current strategy $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ in the i-th iteration. Lines 8-9 compute the change in VP's profit between the current and previous iterations. If the change is less than the threshold φ, the algorithm is considered converged and exits the iteration.

F. Peak-seeking Pricing Optimization for MNO

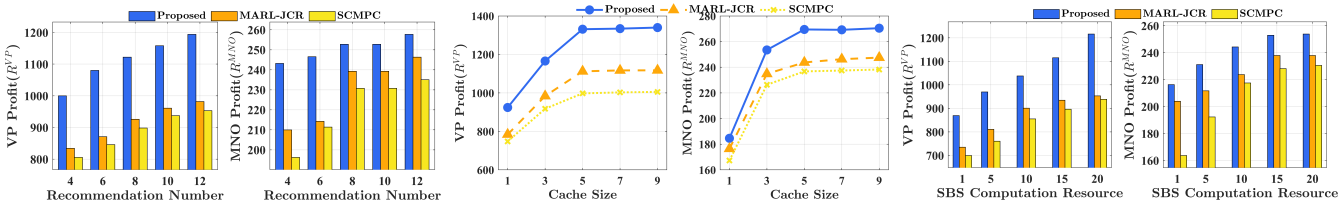
Similar to the findings in [11], the MNO's profit initially increases as the basic cache-leasing price δ increases. This occurs because when δ is low, the VP deploys more high-bitrate videos, taking advantage of the lower leasing costs to boost profits. SBSs' storage capacity is a key constraint on the MNO's profit. As δ rises, the VP adjusts its strategy by deploying fewer low-bitrate videos, thereby reducing the storage space occupied. Consequently, SBSs' storage capacity ceases to be a limiting factor for the MNO's profit. Once δ

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.



(a) SBS cache capacity. (b) Revenue Incentive Factor. (c) MU population size.

Fig. 2: Profit of VP and MNO under varying SBS cache capacities, revenue incentive factors, and MU population sizes.



(a) Recommendation number. (b) SBS cache capacity. (c) SBS computation resource.

Fig. 3: Profit of VP and MNO under varying recommendation numbers, SBS cache capacities, and SBS computation resources.

increases to a certain value, the MNO's profit starts to decline with further increases in δ. This is because, as δ increases, the VP's leased storage space continuously decreases. This trend is verified in Section IV-B. Based on this analysis, we propose a peak-seeking pricing strategy to maximize MNO profit. The strategy begins with δ set to 0 and increases δ incrementally by a step size of $\Delta\delta$ until the MNO's profit starts to decrease or δ reaches its maximum value.

The pseudo-code of the peak-seeking pricing strategy is shown in **Algorithm 5**. Line 1 initializes the price adjustment step size $\Delta\delta$ and line 2 calculates the maximum number of iterations $\mathcal{I}^{\mathrm{MNO}}$. Lines 3-8 iteratively increase the basic cache-leasing price δ to obtain the optimal price δ^* that maximizes MNO's profit R^{MNO}. Specifically, in the i-th iteration, we set δ to $(i-1)\Delta\delta$ to ensure that δ starts from 0 and increases in step size of $\Delta\delta$ (line 4). Line 5 calls **Algorithm 4** to obtain the optimal strategy $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ for the VP under the current basic cache-leasing price δ. MNO's profit $R^{\mathrm{MNO}}(i)$ in the i-th iteration is calculated by Eq. (15) in line 6. The iteration is exited when MNO's profit starts to decrease (lines 7-8). Lines 9-12 determine the optimal basic cache-leasing price δ^* and the corresponding MNO's profit $R^{\mathrm{MNO}*}$. If the iteration stops due to a decline in MNO's profit, then the optimal basic cache-leasing price is the price in the previous iteration (lines 9-10). Otherwise, the optimal basic cache-leasing price is the one from the current iteration (lines 11-12).

IV. Simulation Experiment

A. Experiment Setup

In our experiments, the edge caching system covers a 1km radius with 6 uniformly random SBSs. Each SBS has 3 neighbor links for cooperative caching, and a 200m service range with MUs uniformly distributed. SBS cache capacity ranges from $[1, 9]$GB, CPU frequency $[1, 20]$GHz, and the total compute budget 100Gcycles [18]. Videos have 5 bitrate versions $[15, 5, 2.5, 1.1, 0.7]$Mbps, durations between $[60, 200]$s, and transcoding density 5 cycles/bit [29]. Recommendation and soft-hit thresholds are set from $[0, 0.01]$ and $[0, 0.4]$, respectively. In the economic model, basic MU payment is 3, the price gradient is 1, MNO cache pricing lies in $[0, 10]$ [11], and reward–penalty coefficient varies within $[0, 1]$. Maximum request latency, transmission rate between SBSs, cloud backhaul rate are varied in the range of $[0.8, 1.5]$s, $[10, 40]$Gbps, and $[4, 8]$Gbps, respectively [8]. 4 MU types—quality-oriented, price-sensitive, balanced, and neutral—are modeled, with bitrate preferences drawn from Gaussian or uniform distributions.

To comprehensively evaluate our scheme, we conduct experiments under various configurations. We first examine the impact of SBS cache capacity, the revenue incentive factor, and the MU population size. The SBS cache capacity is set at three levels: small, medium, and large. The revenue incentive factor is set to 0.2, 0.5, and 0.8, respectively. The MU population size ranges from 30 to 50. We then compare our scheme with two SOTA methods, and perform ablation studies to assess each component's contribution. Performance metrics are MNO profit and VP profit.

B. Impact of Key Parameters on System Profit

The impact of SBS cache capacity on the profit of both MNO and VP is illustrated in Fig. 2(a). MNO profit first rises with capacity due to enhanced service capability and higher leasing income, then plateaus or declines as maintenance costs dominate. This non-linear relationship highlights the importance of selecting an optimal cache size that balances infrastructure investment with operational gains. Conversely, VP profit generally increases with larger cache capacity, as more content can be stored locally, reducing backhaul costs

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

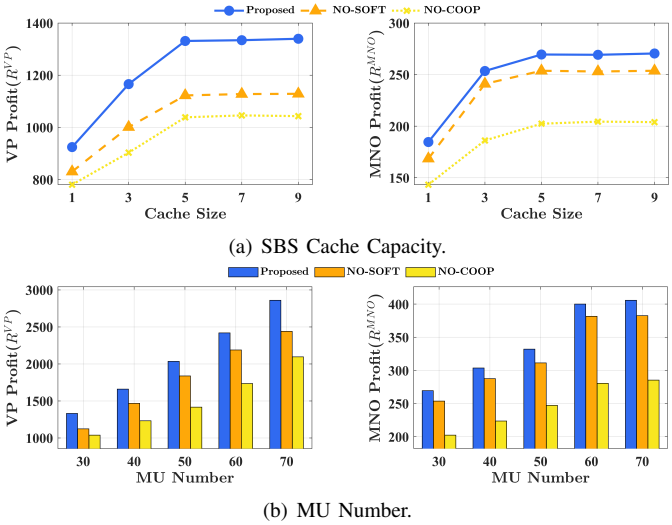
and improving MU experience, although the marginal benefit diminishes at higher capacity levels. Fig. 2(b) illustrates the effect of revenue incentive factor ϵ. A moderate ϵ strikes a balance between encouraging timely delivery and maintaining VP profitability, whereas an excessively high ϵ may reduce overall profit due to large penalties, emphasizing the need for careful calibration of incentive mechanisms. Fig. 2(c) presents scalability with MU population size. Both MNO and VP profit grow with more MUs due to higher demand and cache utilization, but the growth rate tapers under heavier load because of resource contention and request diversity, highlighting the need for adaptive caching and recommendation strategies. Overall, cache capacity, incentive factor, and MU population significantly impact profit, and as δ increases, VP's leased storage decreases. The most balanced profit occurs with a large cache capacity, $\epsilon = 0.8$, and MU population 50, which are adopted in subsequent experiments.

C. Comparative Experiment

We compare the performance of our scheme with two SOTA methods in increasing MNO and VP profit under varying settings of recommendation video number, SBS cache size, and SBS computation resource with 30 MUs and 50 videos.

- **MARL-JCR** [30]: It is a multi-agent reinforcement-learning framework designed to jointly optimize content recommendation and cooperative caching. The framework uses a fully decentralized agent-training method to learn adaptive policies and maximize collective profit.
- **SCMPC** [11]: It is a Stackelberg game-theoretic framework for jointly optimizing video pricing and edge cache placement. The framework adopts a split cache strategy and an iterative gradient ascent algorithm to achieve the Stackelberg Equilibrium.

Fig. 3(a) compares the profit achieved by three approaches under varying video recommendation numbers. The proposed scheme achieves up to 23.99% and 27.69% VP profit improvements over MARL-JCR and SCMPC, respectively, when 8-10 videos are recommended per MU. It also yields up to 15.83% and 23.93% MNO profit improvements. This is because our recommendation mechanism proactively shapes MU requests toward cached content, increasing hit rates and profit while avoiding information overload. Fig. 3(b) examines the performance of three approaches under different SBS cache capacities. The proposed scheme consistently achieves the highest profit for both MNO and VP across all settings. It improves VP profit by up to 19.88% over MARL-JCR and 33.43% over SCMPC, and MNO profit by 10.55% and 13.82%, respectively. The reason is that our scheme jointly optimizes video recommendation and edge caching to efficiently allocate limited cache resources across video versions and capacity constraints. Fig. 3(c) evaluates their performance under varying SBS computational resources. Our scheme achieves VP profit gains up to 27.60% and 29.52%, and MNO profit gains up to 9.20% and 32.18% over MARL-JCR and SCMPC, respectively. This superiority stems from our adaptive request scheduling that orchestrates computation resources to balance service quality and profit, showing strong adaptability under varied loads.



(a) SBS Cache Capacity.

(b) MU Number.

Fig. 4: Profit of VP and MNO under varying SBS cache capacities and MU numbers.

D. Ablation Experiment

The performance of the complete proposed scheme is compared with two degraded versions: one that removes the soft-hit policy (NO-SOFT) and another that disables cooperative caching among neighboring SBSs (NO-COOP). Fig. 4(a) shows that our scheme consistently outperforms NO-SOFT and NO-COOP across cache sizes. Removing soft-hit reduces VP and MNO profit by up to 18.6% and 9.6%, respectively, as more requests fall back to costly backhaul. Disabling cooperation causes even larger losses, up to 29.1% for VP and 36.2% for MNO, due to reduced caching efficiency. Notably, soft-hit primarily benefits VP by enabling alternative content delivery, while cooperative caching mainly boosts MNO profit by improving cache utilization and promoting more leasing.

Fig. 4(b) compares the three schemes under varying MU scales (30–70). Both profits increase with MU population, with our scheme consistently outperforming baselines. Against NO-SOFT, VP and MNO gains reach 18.6% and 6.7%, respectively; against NO-COOP, the gains reach 43.8% and 42.6%. Larger MU scales increase request overlap for popular content, amplifying cache reuse. Soft-hit leverages this by serving similar requests with cached alternatives, boosting VP flexibility without extra storage. Meanwhile, cooperative caching addresses diversity via cross-SBS sharing, improving overall hit rates. The synergy of both mechanisms optimizes resource efficiency and economic benefits across scales.

V. Conclusion

This paper studies the joint optimization of recommendation and cooperative caching for multi-bitrate VoD to maximize the profits of both MNO and VP. To tackle their intertwined interests, the heterogeneity and scarcity of edge resources, and the diversity of user preferences, we propose BIS, a Stackelberg game-driven framework that combines layered

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.

optimization with alternating iterations. BIS models profit-sharing as a leader–follower game for short-term revenue coordination under economic incentives. On the VP side, a soft-cache-hit mechanism and neighbor cooperation enhance cache utilization. The MNO searches for optimal cache price with a fixed step size, while the VP optimizes video recommendation, cache placement, and delivery under that price. To handle the non-convexity and discrete variables of the VP subproblem, we decompose it via BCD and solve each block alternately, using continuous relaxation and rounding techniques for high-quality discrete solutions. Extensive simulation experiments validate the efficacy of our proposed framework.

ACKNOWLEDGE

This work was supported in part by the National Natural Science Foundation of China under Grant 62572244 and Grant 62302216, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20230913 and Grant BK20230912, in part by the Fundamental Research Funds for the Central Universities under Grant 30924010815, and in part by the open research fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, under Grant ESSCKF2024-07.

REFERENCES

[1] "Ericsson mobility report," accessed 2025. [Online]. Available: https://www.ericsson.com/49e9b6/assets/local/reports-papers/mobility-report/documents/2025/ericsson-mobility-report-june-2025.pdf

[2] C. Gao and T. Braun, "Dual-engine intelligent caching: A joint optimization framework for 360° mobile vr video edge caching," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 10, pp. 3532–3547, 2025.

[3] W. Liu, H. Zhang, H. Ding, Z. Yu, and D. Yuan, "Qoe-aware collaborative edge caching and computing for adaptive video streaming," *IEEE Transactions on Wireless Communications*, vol. 23, no. 6, pp. 6453–6466, 2024.

[4] Y. Li, H. Ma, L. Wang, S. Mao, and G. Wang, "Optimized content caching and user association for edge computing in densely deployed heterogeneous networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2130–2142, 2022.

[5] L. Li, D. Shi, R. Hou, R. Chen, B. Lin, and M. Pan, "Energy-efficient proactive caching for adaptive video streaming via data-driven optimization," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5549–5561, 2020.

[6] H. Wu, H. Lu, F. Wu, and C. W. Chen, "Energy and delay optimization for cache-enabled dense small cell networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7663–7678, 2020.

[7] F. Wang, F. Wang, J. Liu, R. Shea, and L. Sun, "Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach," in *IEEE International Conference on Computer Communications*, 2020, pp. 2499–2508.

[8] T. X. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 1965–1978, 2019.

[9] C. Li, Y. Zhang, and Y. Luo, "A federated learning-based edge caching approach for mobile edge computing-enabled intelligent connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3360–3369, 2023.

[10] Z. Hu, C. Fang, Z. Wang, S.-M. Tseng, and M. Dong, "Many-objective optimization-based content popularity prediction for cache-assisted cloud–edge–end collaborative iot networks," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1190–1200, 2024.

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

[12] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient video pricing and caching in heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8744–8751, 2016.

[13] C. Fang, C. Liu, Z. Wang, Y. Sun, W. Ni, P. Li, and S. Guo, "Cache-assisted content delivery in wireless networks: A new game theoretic model," *Information Systems Journal*, vol. 15, no. 2, pp. 2653–2664, 2021.

[14] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *IEEE International Conference on Computer Communications*, 2017, pp. 1–9.

[15] Y. Zhao, Z. Yu, and D. Yuan, "Caching with personalized and incumbent-aware recommendation: Modeling and optimization," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9595–9613, 2024.

[16] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1300–1313, 2018.

[17] C. Sun, X. Li, J. Wen, X. Wang, Z. Han, and V. C. M. Leung, "Federated deep reinforcement learning for recommendation-enabled edge caching in mobile edge-cloud computing networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 690–705, 2023.

[18] W. Liu, H. Zhang, H. Ding, Z. Yu, and D. Yuan, "Recommendation based video caching and transcoding in mobile edge networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 5, pp. 6572–6583, 2024.

[19] M. Costantini and T. Spyropoulos, "Impact of popular content relational structure on joint caching and recommendation policies," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2020, pp. 1–8.

[20] P. Cong, J. Zhou, M. Chen, and T. Wei, "Personality-guided cloud pricing via reinforcement learning," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 925–943, 2020.

[21] Y. Zeng, J. Zhou, B. Ye, Z. Qu, S. Guo, T. Gong, and P. Li, "Expertdrl: Request dispatching and instance configuration for serverless edge inference with foundation models," *IEEE Transactions on Mobile Computing*, vol. 24, no. 9, pp. 8089–8104, 2025.

[22] J. Ke, J. Zhou, D. Meng, Y. Zeng, Y. Shi, X. Qu, and S. Guo, "Jcsrc: Joint client selection and resource configuration for energy-efficient multi-task federated learning," *IEEE Transactions on Computers*, vol. 74, no. 12, pp. 4094–4108, 2025.

[23] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Jointly optimizing content caching and recommendations in small cell networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 125–138, 2019.

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

[25] Y. Fu, Y. Zhang, Q. Zhu, M. Chen, and T. Q. S. Quek, "Joint content caching, recommendation, and transmission optimization for next generation multiple access networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1600–1614, 2022.

[26] Q. Lu, C. Li, J. Zou, K. Tang, Q. Wang, and H. Xiong, "Transcoding-enabled edge caching and delivery for tile-based adaptive 360-degree video streaming," in *IEEE International Conference on Visual Communications and Image Processing*, 2019, pp. 1–4.

[27] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, Dec. 2015.

[28] J. Bai, S. Zhu, and H. Ji, "Blockchain based decentralized and proactive caching strategy in mobile edge computing environment," *Sensors*, vol. 24, no. 7, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/7/2279

[29] E. Ghabashneh and S. Rao, "Exploring the interplay between cdn caching and video streaming performance," in *IEEE International Conference on Computer Communications*, 2020, pp. 516–525.

[30] X. Zhou, Z. Ke, and T. Qiu, "Recommendation-driven multi-cell cooperative caching: A multi-agent reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4764–4776, 2024.

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on November 05,2026 at 12:47:51 UTC from IEEE Xplore. Restrictions apply.