Received 23 June 2020; revised 8 August 2020; accepted 19 August 2020. Date of publication 21 August 2020; date of current version 11 September 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.3018681

Joint Video Caching and Processing for Multi-Bitrate Videos in Ultra-Dense HetNets

TICAO ZHANG (Graduate Student Member, IEEE), AND SHIWEN MAO (Fellow, IEEE)

Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA

CORRESPONDING AUTHOR: S. MAO (e-mail: smao@ieee.org)

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant ECCS-1923717 and Grant CNS-1822055, through the Wireless Engineering Research and Education Center, Auburn University.

ABSTRACT Caching popular videos at the edge has been confirmed as a promising way to support low-latency video transmission and alleviate the backhaul traffic burden. Meanwhile, mobile edge computing (MEC) has also been regarded as an effective solution to meet the 5G low-latency service requirements. In this article, we propose to fully utilize both the storage and computing resources at edge servers to support multiple bitrate video streaming. We design the video caching, processing, and user association models that aim to minimize the average retrieval latency of all users. This problem is modeled as a mixed-integer bilinear problem, which is *NP-hard*. We show that under practical constraints on storage, bandwidth, and processing capacity, the problem does not exhibit sub-modular property and the performance of a greedy algorithm may not be strictly guaranteed. To deal with this challenging problem, we decompose the original problem into a cache placement problem and a user-BS association problem, while still preserving the interplay between the two sub-problems. A linearization and rounding algorithm, including: (i) a greedy rounding proactively caching scheme and (ii) a random-rounding user-BS association scheme, is then proposed, with performance bounds derived. Extensive simulation results show that the proposed scheme can achieve a near-optimal performance under various storage, computing capacity, and downlink bandwidth settings.

INDEX TERMS Video caching, video transcoding, multiple bitrate video, mobile edge computing (MEC), submodularity.

I. INTRODUCTION

Wireless traffic has increased significantly in the past decades due to the rapid development of mobile communication technologies. With the growing success of on-demand video services, video traffic now is dominating the mobile traffic. According to Cisco, global mobile traffic will reach 77.5 exabytes/month by the year 2022 and video traffic will account for 79% of it [1]. Traditional approaches to deal with the dramatic traffic growth is spectrum expansion, network densification, and improving the spectral efficiency or spatial reuse. However, these approaches are approaching their performance limit and sometimes they are too expensive to implement in practice. The rapid growth of video traffic and the emerging new video applications, such as Augmented reality (AR) and Virtual Reality (VR) [2], [3], bring about great challenges to the existing wireless networks.

It has been observed that video on-demand services often exhibit the asynchronous content property [4], such that a few popular files account for a large part of the traffic, which are requested by users at different times. Caching at the mobile network edge can significantly bring contents closer to users, which naturally reduces the retrieval delay [5] and alleviates the backhaul traffic [6]. Furthermore, with the development of edge computing [7], the computing capacity of the edge cloud has drawn researcher's attention recently. It is expected to be an effective solution to meet the low latency demand of context-aware services and applications [8]. Existing work starts to jointly exploit the storage capacity as well as the computing power of edge servers to enable greatly improved experience for users [9]–[14].

This joint design approach is highly promising because caching alone may not be able to meet the fast growing demands of emerging video applications in 5G wireless and beyond. For example, in AR [2], [3], [15], video object classification and recognition task has to be performed first and then the videos are delivered to the user. In multi-viewpoint

360 degree interactive video transmission, the viewing-related features have to be analyzed at the edge first, then the video quality and other video transmission related parameters will be determined [16]. Future communication networks will require not only wireless content caching, but also considerable content processing at the edge.

Modern commercial video streaming services such as Youtube, Netflix, and Apple [17] adopt adaptive bitrate (ABR) video transmission. In ABR video streaming, different versions of the same video are encoded at different bitrates and delivered to users based on the user device and the channel conditions. ABR video streaming has been widely used to improve the user quality of experience (QoE) in wired/wireless networks [18], [19]. Now with the edge computing power being considered, the joint video caching and processing scheme has the potential to further improve the QoE performance. For example, if the requested bitrate video version is not cached, but however, a higher bitrate version is cached at the edge server, then the edge server can leverage its computing capacity to process the video, e.g., via video transcoding, to transform the video to the target bitrate and then deliver it to the requesting user. In this context, the joint video caching and processing has at least two potential benefits: (i) users can obtain videos at the bitrate that is best suited to their requirements as well as their channel conditions; (ii) the remote server does not need to cache all the bitrate versions of the same video at the local servers.

However, this framework also faces many challenges. First of all, due to the storage limitation of edge servers, only limited popular videos can be cached at the edge. There is a tradeoff between caching for high bitrate videos (quality) and caching for different types of videos (diversity). Second, real-time video processing, especially video transcoding, is a computation-intensive task that would consume considerable computing resources. It is challenging to design a proper resource allocation algorithm to fully utilize the *storage* resources as well as the *computing* resources. Moreover, in multi-bitrate video streaming, videos can only be transcoded from a higher bitrate version to a lower bitrate version. The dependency between different versions of the same video in turn incurs additional complexity. Third, future edge servers will be deployed in ultra-dense networks (UDN) [20], [21]. By bringing the access nodes as close as possible to users, a huge access capacity can be provided. However, such densification of deployment also increases the complexity of the network. How to assign different users to BSs, especially in the overlapping region, is also a challenging problem.

In this article, we study the joint video caching, transcoding, and user-BS association problem in an ultra-dense heterogeneous network. The main contribution of this article is summarized as follows:

- We consider the practical issues such as the video cache storage, video processing capability, and the downlink bandwidth constraint, and model the problem as a nonlinear mixed-integer bilinear program, which aims to minimize the average video retrieval delay.
- We prove that when jointly considering practical issues such as the downlink bandwidth as well as the computing capability at SBSs, the sub-modularity property does not strictly hold and a greedy algorithm may not be optimal.
- We propose a linearization and rounding method to solve the joint video caching, transcoding, and user-BS association problem. This algorithm decomposes the original problem into a cache placement problem and a user-BS association problem. A greedy rounding proactively caching scheme and a random-rounding user-BS association scheme are then proposed to find a competitive feasible solution.
- We prove a performance bound by introducing auxiliary variables to the original problem. We also derive tight bounds on the performance gaps between the solution produced by the proposed algorithm and the optimal solution. Simulation results verify that the proposed algorithm performs very close to the optimal solution under different practical constraint settings.

This article is organized as follows. Section II reviews related works. The system model and problem formulation are presented in Section III. We prove the non-submodular property of the problem in Section IV and propose effective algorithms in Section V. Our simulation study and discussions are provided in Section VI. Section VII concludes this article.

II. RELATED WORK

This work is closely related to the prior works on wireless caching and adaptive bitrate video caching. We review these two classes of relevant works in this section.

A. WIRELESS CACHING

In [22], [23], the authors first proposed the idea of *femto-caching*, which caches popular video contents in the finite storage of *helper* nodes. This framework has been proven to have the potential to increase the number of served requests. The *proactive* wireless caching technology was later presented in [24], [25]. A mechanism, whereby files are proactively cached during off-peak hours based on popularity, has been shown to be able to effectively alleviate backhaul congestion. Further, device-to-device (D2D) caching has been studied in [4], [23], [26], [27], where mobile devices act as *helper* nodes to store popular video contents and directly serve the requests from neighboring users. The D2D approach offers significant throughput gains [4] and spectral efficiency improvements [26].

B. ADAPTIVE BITRATE VIDEO CACHING

To account for ABR video streaming, one research thread is focused on scalable video coding (SVC) [28]–[31]. In SVC video encoding, each video is encoded into multiple layers, where the base layer (BL) provides a basic viewing quality and one or more enhancement layers (ELs) enable further improved video experiences. However, SVC was not preferred in industry in the past due to the lack of hardware decoding support. Moreover, decoding of multiple video layers usually consumes too much computing resources for mobile users whose computing power and battery are both limited.

Another theme of research considers the multiple bitrate video transmission. In this framework, multi-bitrate video are generated (i.e., via video transcoding) and stored at edge servers. A user chooses a specific video to download and view. Thus the decoding burden at the user side could be significantly reduced. A collaborative video caching and processing scheme that supports ABR video streaming is proposed in [11]. An efficient heuristic cache placement algorithm and low-complexity user request scheduling algorithms are proposed to minimize the expected average delay of video retrieval.

Different from [11] that aims to minimize the average latency, a joint multi-bitrate video caching and processing model is proposed in [10] from the perspective of economics. This scheme aims to maximize the profit of the video service provider while satisfying users' quality requirements. The problem of QoE-aware multi-bitrate video caching is considered in [12]–[14], [32]. In [32], a compound QoE model is proposed for the delivery of Dynamic Adaptive Streaming over HTTP (DASH) videos over an orthogonal frequency-division multiplexing access (OFDMA) network. Algorithms for user equipment (UE) rate adaptation and BS resource allocation are developed, along with a stochastic model predictive control (SMPC) scheme to achieve high robustness on video rate adaption. A QoE-driven mobile edge caching scheme for ABR video streaming scheme is proposed in [12]. This work jointly considers the rate-distortion (RD) characteristics of videos and the coordination among the edge servers and proposes an efficient caching scheme to minimize the aggregated average video distortion. Recently, a joint video caching, power allocation and user association scheme is proposed in [13] to maximize a QoE-aware throughput. Experiments conducted on real user trace datasets demonstrate the effectiveness of the proposed scheme. The recent work [14] presents a comprehensive study on multiple bitrate rate video caching algorithms under both linear and concave QoE functions. The proposed caching scheme can improve user-perceived QoE for positive strictly increasing QoE functions.

In the field of multiple bitrate video streaming, the most related work to ours is [11] and [14], which both leverage the sub-modular property of the caching scheme and develop a greedy (proactive) caching algorithm. Our work differs from [11] and [14] since we show that the sub-modular property may not hold true when jointly considering the computing and bandwidth constraints. A novel linearization and rounding algorithm is thus proposed to achieve competitive performance.

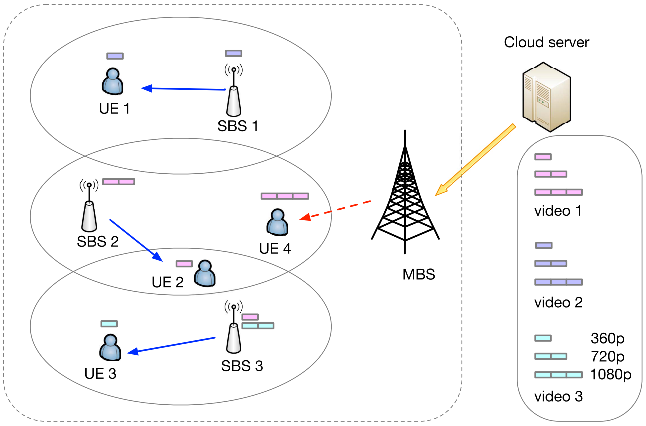


FIGURE 1. A HetNet as a joint video caching and processing system.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. NETWORK MODEL

We consider a system that includes video content providers, wireless service providers, and mobile device users. The mobile users request videos from the video content providers. To reduce the latency and relieve the backhaul traffic burden, the video content providers cache videos on the bases stations (BSs) operated by the wireless service providers. Consider a heterogeneous cellular network (HetNet) as shown in Fig. 1, which consists of one macro-cell base station (MBS), M small-cell base stations (SBSs), and N UEs. Each SBS is associated with a mobile edge server which provides both storage and computational resources.

Let the set of cache-enabled SBS be denoted by $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$ and the set of UEs be denoted by $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, where M and N are the numbers of SBSs and mobile users, respectively. In particular, the MBS is denoted as s_0. The UEs are randomly distributed in the HetNet. Each SBS has a coverage area of radius R_0. We denote $\mathcal{N}_i \subseteq \mathcal{S}$ the set of neighboring SBSs that cover UE u_i. Each of the neighboring SBSs can be a candidate to serve the UE. We also define the neighbor matrix $\Gamma = \{\gamma_{i,j}\}$, whose entry is defined as

$$\gamma_{i,j} = \begin{cases} 1, & \text{if the } u_i - s_j \text{ distance is within } R_0 \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Assume that each SBS has multiple types of resources. First of all, each SBS j has a storage capacity C_j to cache popular video files. Second, each SBS j has a computing capacity W_j (i.e., in the form of the maximum number of CPU cycles). Third, when UEs are demanding videos, each SBS j has a downlink bandwidth capacity B_j.

B. VIDEO MODEL

We consider a video file library $\mathcal{V}$ consisting of P videos and each video is encoded into Q resolutions, denoted by $\mathcal{V} = \{v^{p,q}\}$ and $v^{p,q}$ denotes the pth $(1 \leq p \leq P)$ video with resolution q $(1 \leq p \leq Q)$. The size of video file $v^{p,q}$ is $F^{p,q}$. Suppose $q = 1$ means the lowest video resolution and videos with higher resolutions have larger sizes, then we have $F^{p,1} < F^{p,2} < \cdots < F^{p,Q}$, for all p. For simplicity, we assume that all videos have the same length in time, denoted by T. Thus the average video bitrate is $b^{p,q} = F^{p,q}/T$. Note that our results can be easily extended to the case of different video sizes by partitioning a long video into segments of the same size.

We use a binary variable matrix $\mathbf{Y} = \{y_i^{p,q}\}$ to represent UEs' requests for the videos, where

$$y_i^{p,q} = \begin{cases} 1, & \text{if UE } u_i \text{ presents a request for } v^{p,q} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Assume that the UEs' preferences for different videos follow the Zipf distribution [33] with a skew parameter α. For a specific video, each UE has an equal access probability to the videos with various resolutions, i.e., the probability that an incoming request for a video $v^{p,q}$ is given by

$$\Pr\left(y_i^{p,q} = 1\right) = \frac{1}{Q} \cdot \frac{1/p^{\alpha}}{\sum_{w=1}^{P} 1/w^{\alpha}}. \tag{3}$$

In practice, the UEs' preferences to different videos can be known in advance by learning from history data from a past time period by leveraging sophisticated machine-learning and data-mining algorithm [34]. For ease of analysis, we assume that at each time slot, each UE presents only one video request, i.e.,

$$\sum_{p=1}^{P}\sum_{q=1}^{Q} y_i^{p,q} = 1, \ \forall\, i. \tag{4}$$

This is generally reasonable since a user is less likely to view multiple videos simultaneously. If a UE presents multiple video requests, we can simply regard it as multiple UEs.

C. PROBLEM FORMULATION

Consider the system shown in Fig. 1. The system receives video requests from the UEs in a random manner. As shown in Fig. 1, the target video file is transmitted to the requesting UE in the order of the following three optional ways.

1) *Local Caching:* For UE u_i, if the requested video $v^{p,q}$ is cached in the local storage of a neighboring SBS in $\mathcal{N}_i$ and the SBS has enough bandwidth to deliver the video to the UE, then the UE can directly download the video from the local cache of the SBS (e.g., see UE 1 in Fig. 1).
2) *Mobile Edge Computing:* If the requested bitrate version video is not cached, but however, there is a higher bitrate version in the local cache (e.g., see UE 2 in Fig. 1), or the SBS does not have enough downlink bandwidth due to too many video request (e.g., see UE 3 in Fig. 1), then the SBS will transcode the video to the requested bitrate version and then deliver the transcoded video to the UE.
3) *Backhaul Transmission:* If neither the requested video nor a higher bitrate version is available at the SBS, the UE will request the video file from the MBS via backhaul transmission. Note that this may cause a larger delay due to its long transmission distance and the increased backhaul traffic (e.g., see UE 4 in Fig. 1).

When a UE presents a video request, the system needs to decide how to offer the video service and associate the UE with the corresponding SBS that will serve the UE. To formulate this problem, we introduce two sets of decision making variables: (i) The video caching placement variables, which are denoted as $\mathbf{X} = \{x_j^{p,q}\}$, where

$$x_j^{p,q} = \begin{cases} 1, & \text{if video } v^{p,q} \text{ is cached at SBS } s_j \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

and (ii) the user association variables, which are denoted by $\boldsymbol{\alpha} = \{\alpha_{i,j}^{p,q}\}$, where

$$\alpha_{i,j}^{p,q} = \begin{cases} 1, & \text{if UE } u_i \text{ receives video } v^{p,q} \text{ from SBS } s_j \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

and $\boldsymbol{\beta} = \{\beta_i^{p,q}\}$, where

$$\beta_i^{p,q} = \begin{cases} 1, & \text{if UE } u_i \text{ receives video } v^{p,q} \text{ from the MBS} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The video caching placement and UE association processes need to satisfy several constraints. First, the UE can be served by an SBS if and only if the requested video or a higher bitrate version of the video is cached at the SBS, i.e.,

$$\alpha_{i,j}^{p,q} \le \sum_{q'=q}^{Q} x_j^{p,q'}, \ \forall\, i, j, p, q. \tag{8}$$

Second, at each time slot, each UE's video request must be served, either from a nearby SBS or the MBS, i.e.,

$$y_i^{p,q} \le \beta_i^{p,q} + \sum_{j=1}^{M} \alpha_{i,j}^{p,q}, \ \forall\, i, p, q. \tag{9}$$

Third, the UE can only communicate with a neighboring SBS that is within the communication range or with the MBS, i.e.,

$$\alpha_{i,j}^{p,q} \le \gamma_{i,j}, \ \forall\, i, j, p, q. \tag{10}$$

Forth, the total size of the videos that are placed in the local cache of the SBS should not exceed its storage capacity, i.e.,

$$\sum_{p=1}^{P}\sum_{q=1}^{Q} x_j^{p} F^{p,q} \le C_j, \ \forall\, j. \tag{11}$$

Fifth, if video transcoding is needed, the SBS will convert the video to the target bitrate version. This process will consume additional computational resources. We assume that the basic computing burden to process a video is $w_0^{p,q}$, which may include video feature analysis and the additional content analysis. In addition to basic video processing, the video transcoding incurs consumption of more computing resources. We denote the total computing cost in this part as $w_1^{p,q}$, with $w_1^{p,q} \geq w_0^{p,q}$. Therefore, the overall computing resource constraint is given by

$$\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\alpha_{i,j}^{p,q}\left(x_j^{p,q}w_0^{p,q}+\left(1-x_j^{p,q}\right)w_1^{p,q}\right) \leq W_j,\ \forall j. \tag{12}$$

Finally, the total amount of UE video requests served by SBS j cannot exceed the download capacity of the SBS, i.e.,

$$\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\alpha_{i,j}^{p,q}b^{p,q} \leq B_j,\ \forall j. \tag{13}$$

We aim to design a joint video caching, transcoding, and UE association strategy so that the latency of all UEs can be minimized. Suppose when a UE receives its requested video from the MBS via the backhaul, the delay is t_0; when the UE receives the requested video from a local SBS (either via video transcoding or directly downloading), the corresponding delay is t_1 ($t_0 \gg t_1$) [11], [35]. Then the latency minimization problem can be formulated as follows.

$$\begin{aligned}(\text{P1})\ \min_{\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}}\ & \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\alpha_{i,j}^{p,q}t_1+\beta_i^{p,q}t_0\right)\\ \text{s.t.}\ & \alpha_{i,j}^{p,q},\beta_i^{p,q},x_j^{p,q}\in\{0,1\},\\ & (8)-(13).\end{aligned} \tag{14}$$

IV. PROBLEM ANALYSIS

The decision variables of Problem (P1) are all binary, and the constraint (12) involves the product of decision variables. Therefore, Problem (P1) belongs to the class of nonlinear integer programming. Specifically, this problem is a mixed-integer bilinear problem [36]. This problem is *NP-hard*, since when constraints (12) and (13) are removed, the reduced problem can be generalized to a knapsack problem with a set of knapsack constraints. The reduced problem has been shown to be *NP-hard* (Please refer to [37, Th. 1] for a detailed proof of NP-hardness). It generally requires an exponential complexity to find the optimum with standard optimization solvers, such as MOSEK. In this section, we first study the properties of a reduced problem and develop an effective greedy algorithm. We then focus on the more challenging general problem and derive its non-submodular property.

A. CONVENTIONAL VIDEO CACHING PROBLEM

The conventional video caching problem considers the case where each SBS has infinite computational resources as well as sufficient downlink bandwidth, i.e., constraints (12) and (13) are removed. This is an approximation of the cases when the SBSs have sufficient computation power and downlink bandwidth, or when the load is light. The reduced problem has the following form.

$$\begin{aligned}(\text{P2})\ \min_{\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}}\ & \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\alpha_{i,j}^{p,q}t_1+\beta_i^{p,q}t_0\right)\\ \text{s.t.}\ & \alpha_{i,j}^{p,q},\beta_i^{p,q},x_j^{p,q}\in\{0,1\},\\ & (8)-(11).\end{aligned} \tag{15}$$

This special case has been well studied in the conventional video caching literature, e.g., see [6], [12], when each video has only one bitrate variant. Despite its NP-hard property, some efficient greedy algorithms have been developed. The main idea is to first transform the video caching problem into an equivalent submodular-maximization problem with a set of knapsack constraints, and then develop polynomial-time greedy algorithms.

For the problem of multi-bitrate video caching, we can show that this problem has the submodularity property. To proceed, we first rewrite the objective function as follows.

$$\begin{aligned}D &= \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{p=1}^{M}\sum_{q=1}^{Q}\sum_{j=1}^{M}\alpha_{i,j}^{p,q}t_1+\sum_{p=1}^{P}\sum_{q=1}^{Q}\beta_i^{p,q}t_0\right)\\ &\geq \frac{1}{N}\sum_{i=1}^{N}\left(t_1+(t_0-t_1)\sum_{p=1}^{M}\sum_{q=1}^{Q}\sum_{j=1}^{M}\beta_i^{p,q}\right).\end{aligned} \tag{16}$$

The inequality in (16) comes from constraints (9). As a result, minimizing the average delay is equivalent to minimizing the number of video requests served by the MBS. Problem (P2) is equivalent to Problem (P3), given by

$$\begin{aligned}(\text{P3})\ \max_{\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}}\ & -\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\beta_i^{p,q}\\ \text{s.t.}\ & \alpha_{i,j}^{p,q},\beta_i^{p,q},x_j^{p,q}\in\{0,1\},\\ & (8)-(11).\end{aligned} \tag{17}$$

For a given caching matrix $\mathbf{x}=[x_j^{p,q}]$, suppose the corresponding caching set is denoted as $\mathcal{X}=\{X_j^{p,q}\}$. We can obtain the optimal solution to Problem (P3) as

$$\beta_i^{p,q}=\max\left\{0,\ y_i^{p,q}-\sum_{j\in\mathcal{N}_i}\sum_{q'=q}^{Q}x_j^{p,q'}\right\},\ \forall i. \tag{18}$$

The solution is trivial since the UE will only receive video from the MBS provided that there is no target bitrate video or a higher bitrate video cached in any of its nearby SBS. Denote the objective value to Problem (P3) under the video caching placement $\mathcal{X}$ as $f(\mathcal{X})$. Before proceeding further, we first introduce several basic definitions.

Definition 1: For two caching schemes $\mathcal{X}_1$ and $\mathcal{X}_2 \subseteq \mathcal{X}$, define

$$\Delta_{\mathcal{X}_1}(\mathcal{X}_2)=f(\mathcal{X}_1\cup\mathcal{X}_2)-f(\mathcal{X}_1), \tag{19}$$

i.e., $\Delta_{\mathcal{X}_1}(\mathcal{X}_2)$ is the increment of the total perceived objective value incurred by performing an additional caching scheme $\mathcal{X}_2-\mathcal{X}_1$ over a current caching scheme $\mathcal{X}_1$.

Definition 2: A function $f(\mathcal{X})$ has the sub-modularity property if for any two caching schemes $\mathcal{X}_1 \subseteq \mathcal{X}_2 \subseteq \mathcal{X}$ and any video placement $X_j^{p,q}\in\mathcal{X}$, we have

$$\Delta_{\mathcal{X}_1}\left(X_j^{p,q}\right)\geq\Delta_{\mathcal{X}_2}\left(X_j^{p,q}\right). \tag{20}$$

Proposition 1: The objective function of Problem (P3) is a monotone submodular function over the cache placement set defined by $\mathcal{X} = \{X_j^{p,q} | x_j^{p,q} = 1\}$.

Proof (Monotonicity): Now we consider two video caching schemes, $\mathcal{X}_1 \subseteq \mathcal{X}_2 \subseteq \mathcal{X}$. From (18), it can be easily verified that the possible values of $\beta_i^{p,q}$ corresponding to the video cache set $\mathcal{X}_1$ will be larger than the values corresponding to the video cache set $\mathcal{X}_2$. Since the objective function $f(\mathcal{X})$ is a decreasing function in terms of $\beta_i^{p,q}$, we have $f(\mathcal{X}_2) \geq f(\mathcal{X}_1)$.

Submodularity: Suppose we have two caching schemes, $\mathcal{X}_1 \subseteq \mathcal{X}_2 \subseteq \mathcal{X}$ and an arbitrary new placement $X_j^{p,q} \in \mathcal{X}$. To prove that $f(\mathcal{X})$ is submodular, we need to show that $\Delta_{\mathcal{X}_1}(X_j^{p,q}) \geq \Delta_{\mathcal{X}_2}(X_j^{p,q})$, i.e., the marginal objective function value in Problem (P3) decreases as the size of the cache placement set increases.

We consider the following three cases.

(i) If $X_j^{p,q} \in \mathcal{X}_1$, we have $\Delta_{\mathcal{X}_1}(X_j^{p,q}) = \Delta_{\mathcal{X}_2}(X_j^{p,q}) = 0$. That is, if video $v_j^{p,q}$ is already cached in both $\mathcal{X}_1$ and $\mathcal{X}_2$, the objective value does not increase.

(ii) If $X_j^{p,q} \in \mathcal{X}_2 \setminus \mathcal{X}_1$, it is obvious that $\Delta_{\mathcal{X}_2}(X_j^{p,q}) = 0$. Also, $\Delta_{\mathcal{X}_1}(X_j^{p,q}) \geq 0$, hence $\Delta_{\mathcal{X}_1}(X_j^{p,q}) \geq \Delta_{\mathcal{X}_2}(X_j^{p,q})$. That is, if video $v_j^{p,q}$ is cached in $\mathcal{X}_2$ but not cached in $\mathcal{X}_1$, adding the video to $\mathcal{X}_2$ will not increase the objective value, but additing it to $\mathcal{X}_1$ will.

(iii) If $X_j^{p,q} \in \mathcal{X} \setminus \mathcal{X}_2$, there are two sub-cases to consider. First, if $\exists q' \geq q$, such that $x_j^{p,q'} = 1$ and $X_j^{p,q'} \in \mathcal{X}_1$, (i.e., in the caching strategy $\mathcal{X}_1$, SBS j already caches a higher bitrate version of video $v^{p,q}$), under the assumption that each SBS has infinite video processing capacity and bandwidth, caching a lower version video $v^{p,q}$ at SBS j brings no additional benefit. In this case, $\Delta_{\mathcal{X}_1}(X_j^{p,q}) = \Delta_{\mathcal{X}_2}(X_j^{p,q}) = 0$. Otherwise, caching $v^{p,q}$ in SBS j will enable the neighboring UEs to have access to video $v^{p,q}$ or a lower bitrate version of $v^{p,q}$, which is beneficial in terms of latency reduction. Let $\mathcal{U}_1$ denote the set of UEs under the coverage of SBS j that request to download video $v^{p,q}$ or a lower bitrate version of $v^{p,q}$ from the MBS, and $\mathcal{U}_2$ under caching scheme $\mathcal{X}_2$. Since $\mathcal{X}_1 \subseteq \mathcal{X}_2$, we will have $|\mathcal{U}_2| \leq |\mathcal{U}_1|$. Now we cache video $v^{p,q}$ in SBS j, we will have $\Delta_{\mathcal{X}_1}(X_j^{p,q}) = |\mathcal{U}_1|$ and $\Delta_{\mathcal{X}_2}(X_j^{p,q}) = |\mathcal{U}_2|$. As a result, we have $\Delta_{\mathcal{X}_1}(X_j^{p,q}) \geq \Delta_{\mathcal{X}_2}(X_j^{p,q})$.

According to Definition 2, Problem (P3) is monotone submodular. ■

With the monotone and submodular property, the marginal value of adding a video to the cache placement set will decrease as the cache placement sets grows. An effective way to maximize a monotone sub-modular function is to start with an empty set and at each step, add the element that achieves the highest marginal gain to the set under the constraints. Such a greedy algorithm has been proven to have low complexity and achieve a good approximation guarantee. The complete algorithm is presented in Algorithm 1.

Algorithm 1 Greedy Video Caching Algorithm

```
1:  Input: Cache capacities $\{C_j\}$, encoding bitrate for the
    videos $\{b^{p,q}\}$, video length $T$ ;
2:  Output: Video caching placement $\mathbf{x}$ ;
3:  Initialization $\mathcal{X}_j^{(0)}$ for each SBS as an empty set; Each
    SBS has a copy of the entire video library, i.e., $\mathcal{V}_j^{(0)} = \mathcal{V}$,
    $t = 0$; The caching status for all the SBS is denoted as
    $\mathcal{X}^{(t)} = \{\mathcal{X}_j^{(t)}\}$ and $\mathcal{V}^{(t)} = \{\mathcal{V}_j^{(t)}\}$ ;
4:  while $\mathcal{V}^{(t)} \setminus \mathcal{X}^{(t)} \neq \emptyset$ do
5:      Compute $f(\mathcal{X}^{(t)})$ ;
6:      for $j = 1{:}M$ do
7:          for each video $v^{p,q}$ in $\mathcal{V}_j^{(t)} \setminus \mathcal{X}_j^{(t)}$ do
8:              Compute $f(\mathcal{X}^{(t)} \cup X_j^{p,q})$ ;
9:          end for
10:     end for
11:     $j^*, p^*, q^* \leftarrow \underset{j,p,q}{\text{argmax}}\, \Delta_{\mathcal{X}^{(t)}}(X_j^{p,q})$ ;
12:     $x_{j^*}^{p^*,q^*} = 1$ ;
13:     if $\sum_p \sum_q x_{j^*}^{p,q} F_{p,q} \leq C_j$ then
14:         $\mathcal{X}_{j^*}^{(t+1)} \leftarrow \mathcal{X}_{j^*}^{(t)} \cup \{X_{j^*}^{p^*,q^*}\}$ and $\mathcal{V}_{j^*}^{(t+1)} \leftarrow \mathcal{V}_{j^*}^{(t)}$ ;
15:     else
16:         $\mathcal{X}_{j^*}^{(t+1)} \leftarrow \mathcal{X}_{j^*}^{(t)}$ and $\mathcal{V}_{j^*}^{(t+1)} \leftarrow \mathcal{V}_{j^*}^{(t)} \setminus \{X_{j^*}^{p^*,q^*}\}$ ;
17:         $x_{j^*}^{p^*,q^*} = 0$ ;
18:     end if
19:     if $\Delta_{\mathcal{X}^{(t)}}(X_j^{p,q}) = 0$ then
20:         Break ;
21:     else
22:         $t \leftarrow t + 1$ ;
23:     end if
24: end while
```

Remark 1 (Complexity of the Greedy Algorithm): In Line 5 of Algorithm 1, the objective value is obtained by directly plugging the solution (18) into the objective function or Problem (P3). Overall, Algorithm 1 requires MPQ iterations and in each iteration, there will be MPQ evaluations of the potential marginal gain. As a result, the computational complexity is $\mathcal{O}(M^2P^2Q^2)$. This video placement algorithm does not consider the edge computing capability and bandwidth constraint of each SBS. It is a proactive video caching algorithm that computes the expected average delay. In the cases where the downloading delay for different UEs differ, there is no efficient way to quickly evaluate the objective function. Generally an integer linear programming is required in each iteration. When the number of variables becomes large, the complexity of this algorithm grows quickly.

B. JOINT VIDEO CACHING, TRANSCODING, AND UE ASSOCIATION PROBLEM

In this section, we prove the non-submodular property of the original joint video caching, transcoding and UE association Problem (P1).

Proposition 2: When considering the additional edge computing capacity (12) and the downlink bandwidth constraint (13), the sub-modular property of the objective function of Problem (P1) does not hold.

Proof: Note that the object function of Problem (P3) is derived using constraint (9), which is equivalent to the objective function of Problem (P1). We prove this proposition by constructing a counter-example here.

Consider the simplest case where each video only has one bitrate version and there are $P = 2$ videos in total. The system has $M = 2$ SBSs and $N = 2$ UEs. The UEs are located in the intersection service area of the two SBSs. We assume that the caching capacity and the computing capacity of each SBS are both infinite and only consider the downlink bandwidth constraint in constructing the counter-example. Assume that each SBS can transmit only one video at a time due to the downlink bandwidth constraint.

Now UE 1 presents a request for video 1 and UE 2 present a request for video 2. Consider two caching schemes $\mathcal{A} = \{X_1^1\}$ and $\mathcal{B} = \{X_1^1, X_2^1\}$. In caching scheme $\mathcal{A}$, SBS 1 caches video 1. In caching scheme $\mathcal{B}$, both SBS 1 and SBS 2 cache video 1. Now, we place a video 2 to SBS 1 and compare the marginal objective function increase, i.e., $\Delta_{\mathcal{B}}(X_1^2)$ and $\Delta_{\mathcal{A}}(X_1^2)$.

Under caching scheme $\mathcal{A}$, UE 1's request will be by SBS 1, while UE 2 has to download the video from the MBS. Therefore, the objective function of Problem (P3) (or, Problem (P1)) will be $f(\mathcal{A}) = -1$. For caching scheme $\mathcal{A} \cup X_1^2$, due to the bandwidth constraint of SBS 1, only one of the two requests from UE 1 and UE 2 can be served. The other UE has to download the video from the MBS. As a result, we have $f(\mathcal{A} \cup X_1^2) = -1$ and $\Delta_{\mathcal{A}}(X_1^2) = 0$.

Under caching scheme $\mathcal{B}$, the request of UE 1 can be served by either SBS 1 or SBS 2, while UE 2 still has to download video 2 from the MBS. Thus, we have $f(\mathcal{B}) = -1$. For caching scheme $\mathcal{B} \cup X_1^2$, UE 1 can download video 1 from SBS 2 and UE 2 can download video 2 from SBS 1. In this case, both requests can be served locally, i.e., $f(\mathcal{B} \cup X_1^2) = 0$. We will have $\Delta_{\mathcal{B}}(X_1^2) = 1 > \Delta_{\mathcal{A}}(X_1^2)$. The submodular property will not hold in this situation. ■

Moreover, due to the nonlinear computing constraints in (12), the joint video caching, transcoding, and UE-association problem becomes substantially harder to solve. The greedy algorithm, i.e., Algorithm 1, does not guarantee to provide a near-optimal solution. In the next section, we will present a different methodology that goes beyond the scope of sub-modularity to provide better performance guarantees.

V. LINEARIZATION AND ROUNDING METHOD

In this section, we introduce a linearization and rounding method, which reduces the problem to linear programming and provide a competitive solution to Problem (P1).

A. PERFORMANCE BOUND

Consider the joint video caching, transcoding, and user association problem. Specifically, we observe that the nonlinearity in Problem (P1) comes from the product of $\alpha_{i,j}^{p,q}$ and $x_j^{p,q}$ that appears in (11). We first apply the Reformulation-Linearization Technique (RLT) [38] to derive a relaxed problem. Specifically, we define $z_{i,j}^{p,q} = \alpha_{i,j}^{p,q} x_j^{p,q}$ and $\mathbf{z} = \{z_{i,j}^{p,q}\}$. Then Problem (P1) can be expressed as

$$
\begin{aligned}
\text{(P4)}\quad \min_{\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z}}\quad & \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\alpha_{i,j}^{p,q}t_1 + \beta_i^{p,q}t_0\right) \\
\text{s.t.}\quad & \alpha_{i,j}^{p,q}, \beta_i^{p,q}, x_j^{p,q} \in \{0,1\}, && (21)\\
& z_{i,j}^{p,q} \in \{0,1\},\ \forall\, i,j,p,q && (22)\\
& z_{i,j}^{p,q} \le \alpha_{i,j}^{p,q},\ \forall\, i,j,p,q && (23)\\
& z_{i,j}^{p,q} \le x_j^{p,q},\ \forall\, i,j,p,q && (24)\\
& z_{i,j}^{p,q} \ge x_j^{p,q} + \alpha_{i,j}^{p,q} - 1,\ \forall\, i,j,p,q && (25)\\
& \sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\left(\alpha_{i,j}^{p,q} - z_{i,j}^{p,q}\right)w_1^{p,q} + z_{i,j}^{p,q}w_0^{p,q}\right) \\
& \le W_j,\quad \forall\, j \\
& (8)-(11), \text{and}\ \ (13). && (26)
\end{aligned}
$$

Proposition 3: Problems (P1) and (P4) are equivalent with identical solutions.

Proof: Note that the auxiliary variables are in the form $z_{i,j}^{p,q} = \alpha_{i,j}^{p,q} x_j^{p,q}$. Thus Constraints (12) and (26) are identical. Meanwhile, Problems (P1) and (P4) have identical objective functions. To show that Problems (P1) and (P4) have identical solutions, we only need to show that the additional constraints (22)–(25) are equivalent to the nonlinear constraint $z_{i,j}^{p,q} = \alpha_{i,j}^{p,q} x_j^{p,q}$, where $\alpha_{i,j}^{p,q}$ and $x_j^{p,q}$ are binary decision variables. This can be simply demonstrated by listing all the possible combinations of the values of the two binary variables. ■

The nonlinear integer programming problem (P1) is thus transformed to a linear integer programming problem. The conventional branch-and-cut method incurs an exponential complexity. Moreover, the introduction of the auxiliary variables naturally introduces more indices, which enlarge the search space of the branch-and-cut method. Solving Problem (P4) directly is thus challenging. However, we can easily derive a performance upper bound by relaxing the constraints (2) and (22) to the unit interval [0, 1]. This way, Problem (P4) is simplified to a linear programming problem, which can be easily and optimally solved with a polynomial time complexity by LP solvers such as the simplex method. This results will provide an upper bound for Problem (P1). We can use this theoretical limit to measure the performance of the proposed method described in Section V-B.

B. LINEARIZATION AND ROUNDING ALGORITHM

Solving a large scale nonlinear integer programming problem is always significantly more difficult than solving a linear programming problem. To deal with this issue, our proposed solution is to solve a reduced problem that does not consider the computing constraints first; and then find a feasible solution that meets the computing constraints. This approach is reasonable in practice, since most of the existing systems simply adopt the cache-and-forward scheme; very few systems use the computational resources while caching a video. It is reasonable to assume that the computational resources at the edge are usually redundant at present.

We consider a reduced problem as follows.

$$\text{(P5)}\ \min_{\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\alpha_{i,j}^{p,q}t_1+\beta_i^{p,q}t_0\right) \tag{27}$$

$$\text{s.t.}\ \alpha_{i,j}^{p,q}\in[0,1], \tag{28}$$

$$\beta_i^{p,q}\in[0,1], \tag{29}$$

$$x_j^{p,q}\in[0,1],$$

$$(8)-(11),\text{ and }\ (13), \tag{30}$$

where we remove constraint (12) and replaced constraints (5)–(7) with relaxed constraints (28)–(30). Since both the objective function and the constraints are linear now, we can optimally solve this problem within polynomial time with standard linear programming solvers. Assume that the optimal solution for the video caching and UE-association problem is $\tilde{\mathbf{x}} = \{\tilde{x}_j^{p,q}\}$. To derive a feasible solution to Problem (P1), we need to further round up or down these values to obtain an integer solution. Furthermore, we need to adjust the solution so that the computational capacity constraints are also satisfied.

To deal with these issues, we propose a two-stage rounding algorithm. In the first stage, we round the video caching decision variables to integers so that the storage constraint (11) is satisfied. The algorithm is stated in Algorithm 2. The idea is to solve Problem (P5) first and fix the integer solutions in $\tilde{\mathbf{x}}$. Then we iteratively cache videos by rounding the largest value in $\tilde{\mathbf{x}}$ to 1 until the cache capacity is filled up in each SBS. This process ensures that the storage resources at the SBSs are fully utilized.

Based on the cache placement, in the second stage we consider the joint video processing and UE-BS association problem. Suppose that we obtain the video caching solution $\hat{\mathbf{x}}$ with Algorithm 2. We substitute the solution $\hat{\mathbf{x}}$ back to Problem (P1), and the rest problem becomes finding the optimal UE-BS association given a video placement scheme. This is also a linear integer programming problem. We again first relax the constraints in (6) and (7) to unit intervals and then obtain the factional solutions $\{\tilde{\alpha}_{i,j}^{p,q}\}$ and $\{\tilde{\beta}_j^{p,q}\}$. We then fix the integer elements in $\{\tilde{\alpha}_{i,j}^{p,q}\}$ and guarantee the transmission of these UEs. For the fraction elements in $\{\tilde{\alpha}_{i,j}^{p,q}\}$, we randomly round them to 1 with a probability. The rounding probability depends on the value of the factional element $\{\tilde{\alpha}_{i,j}^{p,q}\}$.

The procedure is summarized in Algorithm 3. Although this is a randomized association algorithm, it is worth noting that this algorithm guarantees all the UE requests. This is confirmed in Line 11 of Algorithm 3. If the UE cannot find a video in its neighboring SBSs, it will download the video from the MBS. This random rounding approach may result in the case that a UE's request is directed to multiple SBSs. To this end, constraints (2)–(11) in Problem (P1) are satisfied. Next we study the remaining constraints (12) and (13).

Algorithm 2 Video Caching Algorithm

```
1: Input: Cache capacity $\{C_j\}$, encoding bitrate for videos $\{b^{p,q}\}$, and video length $T$ ;
2: Output: Cache placement matrix $\hat{\mathbf{x}}$ ;
3: Solve the relaxed linear problem (P5) to obtain the optimal solution $\tilde{\mathbf{x}}$ ;
4: Fixed the integer variables in $\tilde{\mathbf{x}}$ ;
5: For the non-integer variables in $\tilde{\mathbf{x}}$, suppose $\tilde{x}_j^{p,q}$ is the largest one among all the remaining variables in $\tilde{\mathbf{x}}$ that have not been fixed. Round $\tilde{x}_j^{p,q}$ up to 1 ;
6: Repeat Step 5 until constraint (11) is violated ;
7: Output the fixed solution $\hat{\mathbf{x}}$ ;
```

Algorithm 3 The User-BS Association Algorithm

```
1: Input: Cache placement matrix $\hat{\mathbf{x}}$ ;
2: Output: User-BS association matrix $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ ;
3: Relax constraints (6) and (7) in Problem (P1) to unit intervals and solve the relaxed problem with the given $\hat{\mathbf{x}}$ to obtain the optimal solution $\{\tilde{\alpha}_{i,j}^{p,q}\}$ and $\{\tilde{\beta}_j^{p,q}\}$ ;
4: Fix the integer variables in $\{\tilde{\alpha}_{i,j}^{p,q}\}$ and set the corresponding variables in $\{\hat{\alpha}_{i,j}^{p,q}\}$ ;
5: for $i = 1{:}N$ do
6:     for $p = 1{:}P$ do
7:         for $q = 1{:}Q$ do
8:             for $j = 1{:}M$ do
9:                 Set $\hat{\alpha}_{i,j}^{p,q} = 1$ with probability $\tilde{\alpha}_{i,j}^{p,q}$ ;
10:             end for
11:             Set $\hat{\beta}_i^{p,q} = \max\{0,\ y_i^{p,q} - \sum_j \hat{\alpha}_{i,j}^{p,q}\}$ ;
12:         end for
13:     end for
14: end for
15: Output $\{\hat{\alpha}_{i,j}^{p,q}\}$ and $\{\hat{\beta}_i^{p,q}\}$ ;
```

Lemma 1: The solution returned by Algorithm 3 satisfies, in expectation, the computing and downlink bandwidth constraints (12) and (13).

Proof: Note that in Algorithm 3, we round UE-BS association variables to integers with probability $\tilde{\alpha}_{i,j}^{p,q}$. By definition, we have

$$\Pr\left(\hat{\alpha}_{i,j}^{p,q}=1\right)=\tilde{\alpha}_{i,j}^{p,q}\ \text{ and }\ \mathbb{E}\left[\hat{\alpha}_{i,j}^{p,q}\right]=\tilde{\alpha}_{i,j}^{p,q}. \tag{31}$$

We start with the computing constraint first. The expected computation load in SBS j is given by

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{N}\hat{\alpha}_{i,j}^{p,q}\left(\hat{x}_j^{p,q}w_0^{p,q}+\left(1-\hat{x}_j^{p,q}\right)w_1^{p,q}\right)\right]\\
&=\sum_{i=1}^{N}\left[\Pr\left(\hat{\alpha}_{i,j}^{p,q}=1\right)\left(\hat{x}_j^{p,q}w_0^{p,q}+\left(1-\hat{x}_j^{p,q}\right)w_1^{p,q}\right)\right]\\
&=\sum_{i=1}^{N}\tilde{\alpha}_{i,j}^{p,q}\left(\hat{x}_j^{p,q}w_0^{p,q}+\left(1-\hat{x}_j^{p,q}\right)w_1^{p,q}\right)\\
&\le W_j,\ \forall j.
\end{aligned} \tag{32}$$

In (32), the inequality holds true because $\tilde{\alpha}_{i,j}^{p,q}$ are obtained by solving Problem (P1) with the given $\hat{\mathbf{x}}$. Similar inequalities can be shown for the bandwidth constraints, as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\hat{\alpha}_{i,j}^{p,q}b^{p,q}\right] \\
&= \sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\Pr\left(\hat{\alpha}_{i,j}^{p,q}=1\right)b^{p,q} \\
&= \sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\tilde{\alpha}_{i,j}^{p,q}b^{p,q} \qquad (33)\\
&\le B_j,\ \forall j. \qquad (34)
\end{aligned}$$

Therefore, we conclude that the Lemma holds true. ■

In practice, these constraints may still be violated. We have the following theorem that bounds the gap on the downlink bandwidth constraint.

Theorem 1: The downlink bandwidth constraints of SBS j returned by Algorithm 3 will not exceed its bandwidth capacity B_j by a factor of $1+\sqrt{NPQ\ln(NPQ)/(2\mu_j^2)}$ with high probability, where $\mu_j=\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\tilde{\alpha}_{i,j}^{p,q}b^{p,q}$ is the normalized expected bandwidth.

Proof: For a given SBS j, we have already shown that $\mathbb{E}[\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\hat{\alpha}_{i,j}^{p,q}b^{p,q}] = \sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\tilde{\alpha}_{i,j}^{p,q}b^{p,q}$ in (33). By normalizing $b^{p,q}$ and B_j, we can ensure that the variables $\hat{\alpha}_{i,j}^{p,q}b^{p,q}$ fall into the unit interval. Moreover, we already know that $\hat{\alpha}_{i,j}^{p,q}b^{p,q}$ are independent random variables. Now we denote $\mu_j=\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\tilde{\alpha}_{i,j}^{p,q}b^{p,q}$ and apply the Chernoff Bound theorem [39]. We can show that for all $\delta>0$,

$$\Pr\left[\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\hat{\alpha}_{i,j}^{p,q}b^{p,q}\ge(1+\delta)\mu_j\right]\le e^{\frac{-2\delta^2\mu_j^2}{NPQ}}. \quad (35)$$

Note that μ_j should satisfy the constraint $\mu_j\le C_j$, since $\tilde{\alpha}_{i,j}^{p,q}$ is the solution to the relax version of Problem (P1). Therefore, we have

$$\begin{aligned}
&\Pr\left[\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\hat{\alpha}_{i,j}^{p,q}b^{p,q}\ge(1+\delta)C_j\right] \\
&\le \Pr\left[\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\hat{\alpha}_{i,j}^{p,q}b^{p,q}\ge(1+\delta)\mu_j\right] \\
&\le e^{\frac{-2\delta^2\mu_j^2}{NPQ}}. \qquad (36)
\end{aligned}$$

To complete this proof, we need to find a proper value of δ so that the probability that the bandwidth constraint is violated becomes very small. Specifically, we require that

$$e^{\frac{-2\delta^2\mu_j^2}{NPQ}}\le\frac{1}{NPQ}. \quad (37)$$

When the number of video request is very large, the value $\frac{1}{NPQ}$ will be approximately zero. In order for this condition to be true, the value of δ must satisfy

$$\delta\ge\sqrt{\frac{NPQ\ln(NPQ)}{2\mu_j^2}}. \quad (38)$$

We can simply set δ to $\sqrt{NPQ\ln(NPQ)/(2\mu_j^2)}$. Note that in order to apply the Chernoff Bound theorem, we normalize the variables $\hat{\alpha}_{i,j}^{p,q}b^{p,q}$ to unit intervals. Therefore the value of μ_j by definition should fall into the interval $[0, NPQ]$. When NPQ is large, we have $\ln(NPQ)\ll NPQ$. The value of δ depends on the normalized expected bandwidth μ_j. ■

With a similar approach, we can prove the following theorem on the gap on the computation capacity constraint. The proof is omitted for brevity.

Theorem 2: The computing constraint of SBS j returned by Algorithm 3 will not exceed its computation capacity W_j by a factor of $1+\sqrt{NPQ\ln(NPQ)/(2\lambda_j^2)}$ with high probability, where $\lambda_j=\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\tilde{\alpha}_{i,j}^{p,q}(\hat{x}_j^{p,q}w_0^{p,q}+(1-\hat{x}_j^{p,q})w_1^{p,q})$ is the normalized computation load returned by Algorithm 3.

The fractional solution is optimal when given a known cache placement strategy. The rounding procedure in practice will cause a performance loss. The following theorem shows that the expected performance gap, caused by the rounding procedure, can be made zero as long as no UEs are over-served (i.e., a UE's request is served by multiple SBSs).

Theorem 3: The objective value returned by Algorithm 3 is equal to that of the optimal fractional solution in expectation, as long as the UEs are not over-served.

Proof: From constraints (7) and (8), we have

$$\tilde{\beta}_i^{p,q}=\max\left\{0,\ y_i^{p,q}-\sum_j\tilde{\alpha}_{i,j}^{p,q}\right\}. \quad (39)$$

From the rounding step in Line 11 of Algorithm 3, we have

$$\mathbb{E}\left[\hat{\beta}_i^{p,q}\right]=\mathbb{E}\left[\max\left\{0,\ y_i^{p,q}-\sum_j\hat{\alpha}_{i,j}^{p,q}\right\}\right]. \quad (40)$$

Jensen's inequality states that if $\phi(x)$ is a convex function where $x\in\mathcal{D}$, we have $\phi(\mathbb{E}[x])\le\mathbb{E}[\phi(x)]$. In particular, if we choose

$$\phi(x)=\max\{0,\ x\}, \quad (41)$$

we can show that $\phi(x)$ is convex since $\phi(x)$ is the point-wise maximum of two convex functions. Therefore, we have

$$\max\{0,\ \mathbb{E}[x]\}\le\mathbb{E}[\max\{0,\ x\}]. \quad (42)$$

The equality in (42) holds when $x\ge0$, for all $x\in\mathcal{D}$, or when $x\le0$, for all $x\in\mathcal{D}$.

For Problem (P5), we have

$$\mathbb{E}\left[\hat{\beta}_i^{p,q}\right]\ge\max\left\{0,\ \mathbb{E}\left[y_i^{p,q}-\sum_{j=1}^{M}\hat{\alpha}_{i,j}^{p,q}\right]\right\}=\tilde{\beta}_i^{p,q}. \quad (43)$$

The expectation of the objective value of Problem (P1) is

$$\begin{aligned}&\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\hat{\alpha}_{i,j}^{p,q}t_1+\hat{\beta}_i^{p,q}t_0\right)\right]\\ &=\frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\mathbb{E}\left[\hat{\alpha}_{i,j}^{p,q}\right]t_1+\mathbb{E}\left[\hat{\beta}_i^{p,q}\right]t_0\right)\\ &\geq\frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(\sum_{j=1}^{M}\tilde{\alpha}_{i,j}^{p,q}t_1+\tilde{\beta}_i^{p,q}t_0\right).\end{aligned}\tag{44}$$

The inequality in (44) is due to (31) and (43), which suggest that expected objective value returned by Algorithm 3 will be slightly larger than the optimal objective value. To eliminate the performance gap, we have to make sure that the equality holds in (43), i.e., we have to ensure that

$$y_i^{p,q}-\sum_{j=1}^{M}\hat{\alpha}_{i,j}^{p,q}\geq 0,\ \forall\, i,j,p,q.\tag{45}$$

Recall that both $y_i^{p,q}$ and $\hat{\alpha}_{i,j}^{p,q}$ are binary variables, this means we need to make sure that for $y_i^{p,q}$, at most one $\hat{\alpha}_{i,j}^{p,q}$ is set to 1 for any j. In other words, when a UE presents a video request, we need to avoid the situation where this request is routed to multiple neighboring SBSs (i.e., over-serving the UE). When no UEs are over-served, the performance gap between the expected objective value returned by the randomized rounding and that of the optimal fractional solution will be zero. ■

In summary, we have shown so far that

1) The proposed method is optimal *in expectation* as long as no UEs are over-served.
2) The storage capacity is strictly satisfied.
3) All UEs' requests can be met either from neighboring SBSs or the MBS.
4) The bandwidth and computing constraints of each SBS will not be violated by a known factor with high probability.

In practice, there may still be a chance that the computing/bandwidth constraints are violated. To handle this issue, we need to construct a feasible solution so that the computing/bandwidth constraints are always strictly met. To ensure this, we can make a slight modification to Algorithm 3 by adding one more decision below Line 9. If any of the computing power or the bandwidth constraints in SBS j is violated, we set the corresponding $\hat{\alpha}_{i,j}^{p,q}$ to zero. Note that for one specific video request, there is a possibility that the video request is directed to multiple SBSs as given by Algorithm 3. In this case, there will be a performance degradation according to Theorem 3. To handle this issue, the UE can just randomly pick one SBS, and the other SBS(s) will have more resources to serve other UEs.

Despite these practical issues and our fixes, we will show numerically that the obtained conservative solution performs very close to the optimal solution under realistic parameter settings. It is also worth noting that the proposed method only requires solving relaxed linear programming problems, hence this algorithm has a polynomial complexity.

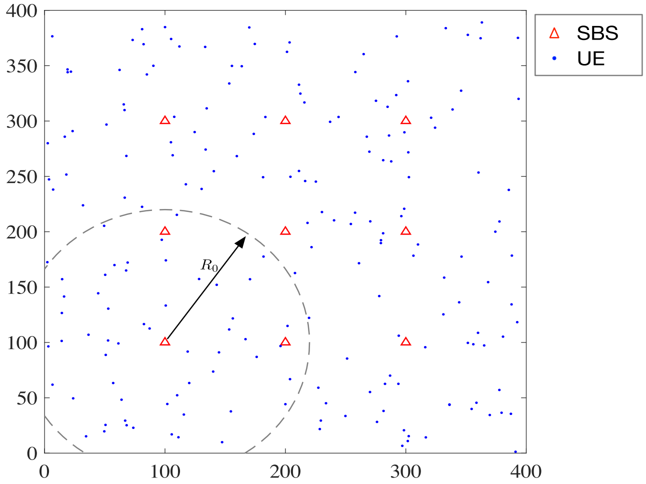


FIGURE 2. Simulation network setup.

VI. SIMULATION STUDY

We simulate an ultra dense HetNet that covers a square area of 400m × 400m with $M = 9$ SBSs and 1 MBS as shown in Fig. 2. The $N = 200$ UEs can connect with an SBS within 120m and all the UEs can connect to the MBS. The UEs are distributed uniformly and independently as shown in Fig. 2.

The total number of videos is $P = 100$. The bitrate in each video is set according to YouTube's encoding policy. Specifically, we choose 4 levels for simplicity, i.e., 1000kbps, 2500kbps, 5000kbps, and 10Mbps [40]. Each video is of the same length, which is $T = 120$ min. We assume that for the same video if video transcoding is needed, the required CPU cycles is uniformly distributed between [0.5, 0.7] GHz; when video transcoding is not needed, the required CPU cycles are uniformly distributed within [0.1, 0.3] GHz. The transmission delay between the UE and the SBS is set as 5ms and from the remote MBS is 100ms. For each SBS, the storage capacity is set to 60 GB, the computation capacity is set to 10 GHz, and the downlink bandwidth is set to 100Mbps. The parameter settings are summarized in Table 1. We compare the proposed method with the following three benchmarks:

1) *Performance bound:* the performance bound is obtained by solving the linear relaxation of Problem (P3) as in Section V-A.
2) *Greedy algorithm:* The greedy algorithm is introduced in Section III. In the first stage, we place videos to the SBSs in a greedy manner to reduce the overall latency, until all the SBS storage is full (while neglecting the bandwidth and computation constraints). In the second stage, given the storage, computing and bandwidth constraints, we associate the UEs to SBSs with Algorithm 3.
3) *Random caching:* Videos are randomly cached at SBSs until their storage is filled up. Algorithm 3 is then used to associate UEs with SBSs in the second stage.

TABLE 1. Simulation parameter setting

Parameter	Value
M	9
R_0	120m
N	200
P	100
$b^{p,q}$	1000kbps, 2500kbps, 5000kbps, 10Mbps
T	120 min
$w_1^{p,q}$	[0.5,0.7] GHZ
$w_0^{p,q}$	[0.1,0.3] GHZ
C_j	60 GB
B_j	100 Mbps
W_j	10 GHz
t_1	5 ms
t_0	100 ms

We use the following metrics in the performance evaluation: (i) *cache hit ratio*, (ii) *average retrieval delay*, and (iii) *external backhaul traffic* load. If the video with the requested bitrate is cached at the UE's associated SBS, the request has an *exact hit*. Otherwise, if video transcoding is needed, we have a *soft hit*. If the video has to be retrieved from the MBS, this would incur additional backhaul traffic. The cache hit ratio is defined as

$$P_{\text{hit}} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{P}\sum_{q=1}^{Q}\alpha_{i,j}^{p,q}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{P}\sum_{q=1}^{Q}\alpha_{i,j}^{p,q} + \sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\beta_{i}^{p,q}}, \tag{46}$$

and the backhaul traffic is computed as

$$T_{\text{backhaul}} = \sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q}\beta_i^{p,q} b^{p,q}. \tag{47}$$

A. IMPACT OF CACHE CAPACITY

We compare the performance of the proposed scheme with the benchmark algorithms under different cache capacities. As a reference, the total size of all the video files is $120 \times 60 \times 18.5 \times 100$ Mbit, which is 1, 665GB. We vary the storage capacity of each SBS from 10GB to 140GB, which is from 0.6% to 8.4% of the total video library size.

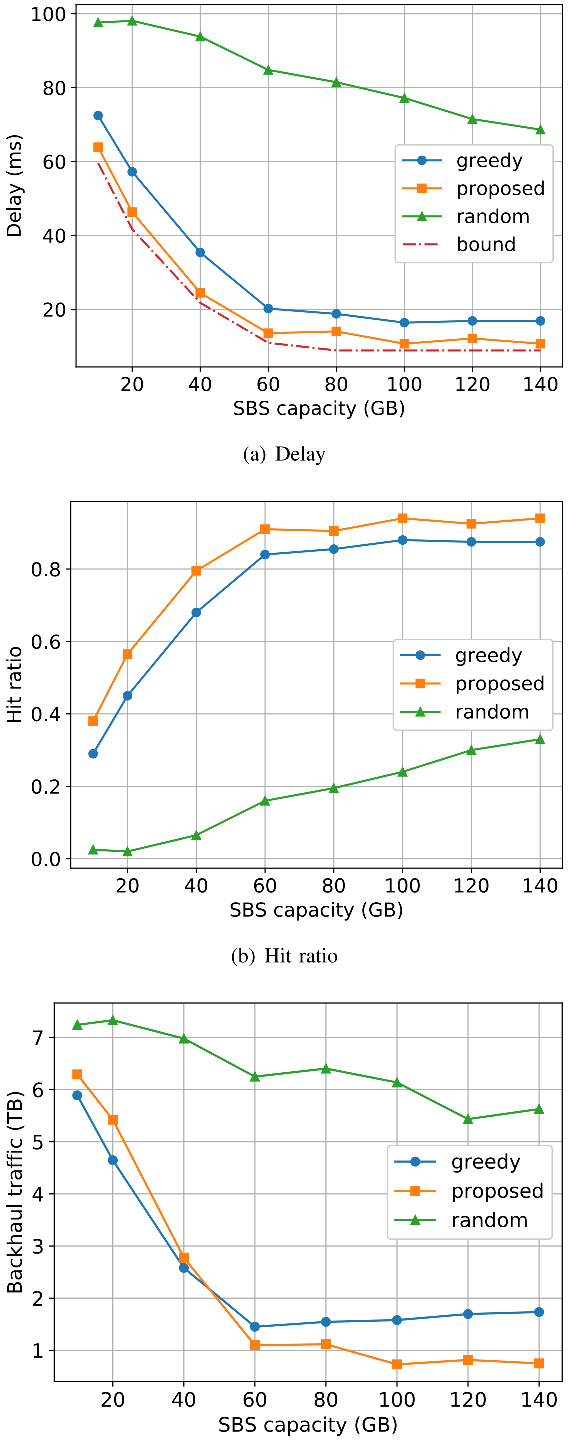
The comparisons of delay, cache hit ratio, and backhaul traffic are presented in Fig. 3. In Fig. 3(a), we find that increasing the storage capacity at the SBS can effectively reduce the transmission delay. The proposed algorithm always outperforms the greedy algorithm and the random scheme in terms of delay performance. This is because the proposed algorithm jointly considers the downlink bandwidth requirement, the layout of the network, users' preferences for the videos, and the storage capacity of each SBS. Moreover, the performance gap between the proposed algorithm and the performance bound is very small, which means the linear relaxation and rounding process is feasible in practice. Fig. 3(b) shows the hit ratio performance, as can be seen the proposed algorithm also achieves the highest hit ratio. The video can be either fetched from the local SBS or through video transcoding. The backhaul traffic can thus be significantly reduced. This is confirmed in Fig. 3(c). Note that, the objective of the proposed method is to minimize the average transmission delay of all the UEs, hence we do not provide the performance bound for the cache hit ratio and the backhaul traffic. In all the three figures, both the proposed algorithm and the greedy algorithm outperform the random algorithm with large gains.

B. IMPACT OF COMPUTING CAPACITY

The impact of the computing capacity of each SBS on the video transmission delay performance is shown in Fig. 4(a). We fix the SBS storage capacity to be 60Gb and the downlink bandwidth to be 100Mbps. We find that with the increase of the SBS computing capacity, the video transmission delay decreases. When the computing capacity is sufficiently high (e.g., over 8GHz), the performance does not improve significantly anymore. Beyond this point, the video storage capacity and downlink bandwidth become the bottleneck to limit the system performance. Recall that, the proposed method transforms the original nonlinear integer programming problem to an integer linear programming problem by not considering the SBS computing power constraint when placing the videos, and only considering the SBS computing power constraint when associating the UEs to SBSs. As a result, the performance gap between the bound and the proposed algorithm is relatively larger when the computing power is moderate (e.g., around 4GHz). However, the proposed method shows an excellent performance when the SBS computing power is small or large. Meanwhile, it always outperforms the greedy and the random algorithms. Fig. 4(b) shows the cache hit ratio performance. By further analyzing the types of cache hits, we find that the increases in the cache hit ratio mainly come from the soft hit type when the SBS computing power goes beyond 4GHz.

C. IMPACT OF DOWNLINK BANDWIDTH

To evaluate the impact of SBS downlink bandwidth, we fix the storage capacity of each SBS to 60GB and the processing capacity to 10GHz. The performance comparison is shown in Fig. 5. In Fig. 5(a), we find that with the increase of the downlink bandwidth of each SBS, the average delay also effectively reduced. However, when the bandwidth is sufficiently large, the transmission delay will not decrease any more. This is because beyond this point, the storage capacity and the processing power will be the performance limiting factor. Also, the performance gap between the proposed method and the performance bound is very small for full range of bandwidth considered. As a comparison, the random caching algorithm generally causes a huge delay. Increasing the downlink transmission bandwidth generally causes little performance improvement for random caching. Fig. 5(b) and Fig. 5(c) depicts the hit ratio performance and the backhaul traffic performance, respectively. As can be seen, when the downlink bandwidth is over 100Mbps, approximate 90% of the video requests can be served by local SBS transmissions and the backhaul traffic is effectively reduced by edge caching.

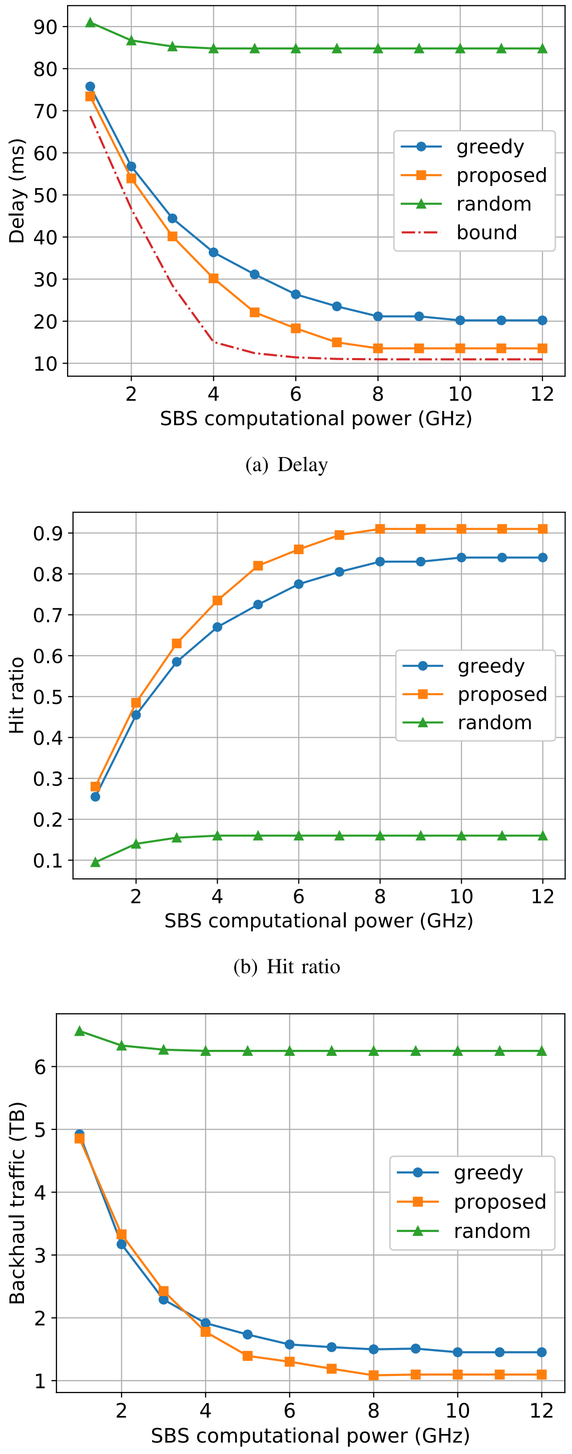


(a) Delay

(b) Hit ratio

(c) Backhaul traffic

FIGURE 3. Impact of the storage capacity at each SBS: $B_j = 100$Mbps and $W_j = 10$GHz, for all j.



(a) Delay

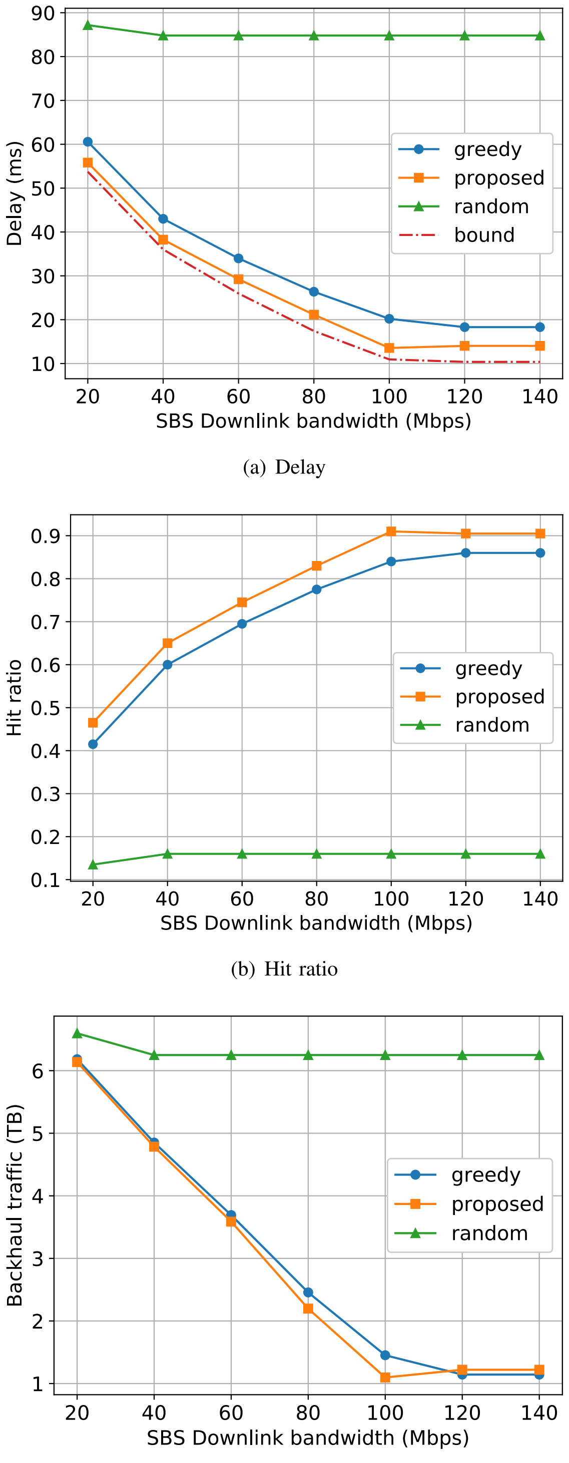
(b) Hit ratio

(c) Backhaul traffic

FIGURE 4. Impact of the computing capacity of each SBS: $C_j = 60$GB and $B_j = 100$Mbps, for all j.

D. PRACTICAL ISSUES

The proposed method makes decisions on how videos are cached, processed, and how UEs are associated with SBSs. This method works for the case when the UE's demands and network topology are fixed or predicted. In practice, the demand may change over time. The timescale is usually at the hours, or even smaller, levels (like the coherence time of wireless channels). The network operator will predict UEs' requests as well as the network topology for the next time period. Based on the prediction, the SBS proactively



(a) Delay

(b) Hit ratio

(c) Backhaul traffic

FIGURE 5. Impact of the downlink bandwidth of each SBS: C_j = 60GB and W_j = 10GHz, for all j.

cache the videos and send the videos to the UEs. Note that there have been prior works that incorporate machine learning models to the prediction of networks traffic and user preferences of videos [9]. Based on these predictions, we believe that the proposed method can significantly reduce the video transmission delay by jointly considering the computing power, storage capacity, and downlink bandwidth of SBSs.

VII. CONCLUSION AND FUTURE WORK

In this article, we investigated the problem of joint video caching, processing, and UE-BS association in an ultra-dense HetNet for the adaptive bitrate video streaming service. In particular, we considered the practical constraints on the SBS, including storage capacity, computing capacity, and the downlink bandwidth. We showed that under this multidimensional, constrained setting, the sub-modular property of the conventional video caching problem does not strictly hold. A linearization and rounding method was proposed to effectively tackle this problem. Simulation results validated that the proposed algorithm achieves a near-optimal performance under different practical constraint settings. Note that the QoE considered in this article is a simplified case where QoE is solely determined by the initial setup latency. It would be interesting to construct a QoE function that considers more factors that affect the video quality, as well as a more accurate model for computing time in future work.

REFERENCES

[1] Cisco Visual Networking Index. (Dec. 2018). *Global Mobile Data Traffic Forecast Update, 2017–2022*. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf

[2] T. Zhang and S. Mao, "Joint power and channel resource optimization in soft multi-view video delivery," *IEEE Access*, vol. 7, pp. 148084–148097, 2019.

[3] J. Dai, Z. Zhang, S. Mao, and D. Liu, "A view synthesis-based 360° VR caching system over MEC-enabled C-RAN," *IEEE Trans. Circuits Syst. Video Technol.*. early access.

[4] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[5] M. Dehghan *et al.*, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proc. IEEE INFOCOM*, Hong Kong, Apr./May 2015, pp. 936–944.

[6] 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.

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[8] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 46–53, Jun. 2013.

[9] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 28–35, Jun. 2018.

[10] Y. Hao, L. Hu, Y. Qian, and M. Chen, "Profit maximization for video caching and processing in edge cloud," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1632–1641, Jul. 2019.

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

[12] 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. 2017.

[13] D. Huang, X. Tao, C. Jiang, S. Cui, and J. Lu, "Trace-driven QoE-aware proactive caching for mobile video streaming in metropolis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 62–76, Jan. 2020.

[14] Z. Qu, B. Ye, B. Tang, S. Guo, S. Lu, and W. Zhuang, "Cooperative caching for multiple bitrate videos in small cell edges," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 288–299, Feb. 2020.

[15] P. Jain, J. Manweiler, and R. Roy Choudhury, "Low bandwidth offload for mobile AR," in *Proc. ACM CoNEXT*, Irvine, CA, USA, Dec. 2016, pp. 237–251.

[16] H. Pang, C. Zhang, F. Wang, J. Liu, and L. Sun, "Towards low latency multi-viewpoint 360° interactive video: A multimodal deep reinforcement learning approach," in *Proc. IEEE INFOCOM*, Paris, France, Apr./May 2019, pp. 991–999.

[17] Apple Inc. *HTTP Live Streaming*. [Online]. Available: https://developer.apple.com/streaming/

[18] T. Stockhammer, "Dynamic adaptive streaming over HTTP—Standards and design principles," in *Proc. ACM MMSys*, San Jose, CA, USA, Feb. 2011, pp. 133–144.

[19] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. ACM MMSys*, San Jose, CA, USA, Feb. 2011, pp. 157–168.

[20] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2522–2545, 4th Quart., 2016.

[21] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[22] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1107–1115.

[23] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[24] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[25] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. 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.

[26] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[27] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2015.

[28] C. Zhan and Z. Wen, "Content cache placement for scalable video in heterogeneous wireless network," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2714–2717, Dec. 2017.

[29] X. Zhang, T. Lv, and S. Yang, "Near-optimal layer placement for scalable videos in cache-enabled small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 9047–9051, Sep. 2018.

[30] Z. Zhang, J. Dai, M. Zeng, D. Liu, and S. Mao, "Scalable video caching for information centric wireless networks," *IEEE Access*, vol. 8, pp. 77272–77284, 2020.

[31] T. Zhang and S. Mao, "Cooperative caching for scalable video transmissions over heterogeneous networks," *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 63–67, Jun. 2019.

[32] K. Xiao, S. Mao, and J. Tugnait, "Robust QoE-driven DASH over OFDMA networks," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 474–486, Feb. 2020.

[33] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of YouTube network traffic at a campus network–measurements, models, and implications," *Computer Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009.

[34] E. Baştuğ *et al.*, "Big data meets TELCOS: A proactive caching perspective," *J. Commun. Netw.*, vol. 17, no. 6, pp. 549–557, Dec. 2015.

[35] X. Li, X. Wang, S. Xiao, and V. C. M. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks," in *Proc. IEEE ICC*, London, U.K., Jun. 2015, pp. 5652–5657.

[36] A. Gupte, S. Ahmed, M. S. Cheon, and S. Dey, "Solving mixed integer bilinear problems using MILP formulations," *SIAM J. Optim.*, vol. 23, no. 2, pp. 721–744, Feb. 2013.

[37] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[38] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, "On path selection and rate allocation for video in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 212–224, Feb. 2009.

[39] M. Mitzenmacher and E. Upfal, *San Francisco, CAProbability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[40] Google. *Choose Live Encoder Settings, Bitrates, and Resolutions—Youtube Help*. [Online]. Available: https://support.google.com/youtube/answer/2853702?hl=en

TICAO ZHANG (Graduate Student Member, IEEE) received the B.E. and M.S. degrees from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Auburn University. His research interests include video coding and communications, machine learning, and optimization and design of wireless multimedia networks. His paper was featured as the IEEE Access Journal's "Article of the Week" in April 2020.

SHIWEN MAO (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Polytechnic University (currently, Tandon School of Engineering, New York University), Brooklyn, NY, USA.

In 2006, he joined the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA, as an Assistant Professor. He held the McWane Professorship from 2012 to 2015. He is currently the Samuel Ginn Endowed Professor, the Director of the Wireless Engineering Research and Education Center, and the Director of the NSF IUCRC FiWIN Center Site, Auburn University. His research interests include wireless networks, multimedia communications, and smart grid.

Dr. Mao received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, the IEEE ComSoc MMTC Distinguished Service Award in 2019, the Auburn University Creative Research and Scholarship Award in 2018, the 2017 IEEE ComSoc ITC Outstanding Service Award, the 2015 IEEE ComSoc TC-CSR Distinguished Service Award, the 2013 IEEE ComSoc MMTC Outstanding Leadership Award, and NSF CAREER Award in 2010. He is a co-recipient of the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE ComSoc MMTC 2018 Best Journal Paper Award, the IEEE ComSoc MMTC 2017 Best Conference Paper Award, the Best Demo Award from IEEE SECON 2017, the Best Paper Awards from IEEE GLOBECOM 2019, 2016, and 2015, IEEE WCNC 2015, and IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a Distinguished Speaker from 2018 to 2021, and was a Distinguished Lecturer of the IEEE Vehicular Technology Society from 2014 to 2018. He was a TPC Co-Chair of IEEE INFOCOM 2018 and is the TPC Vice-Chair of IEEE GLOBECOM 2022. He is an Area Editor of the IEEE Transactions on Wireless Communications, the IEEE Internet of Things Journal, the IEEE Open Journal of the Communications Society, IEEE/CIC China Communications, and *ACM GetMobile*, and an Associate Editor of the IEEE Transactions on Network Science and Engineering, the IEEE Transactions on Mobile Computing, IEEE Multimedia, and IEEE Networking Letters. He is a member of the ACM.