# QoE-Driven Mobile Edge Caching Placement for Adaptive Video Streaming

Chenglin Li, *Member, IEEE,* Laura Toni, *Member, IEEE,* Junni Zou, *Member, IEEE,* Hongkai Xiong, *Senior Member, IEEE,* and Pascal Frossard, *Senior Member, IEEE*

***Abstract*—Caching at mobile edge servers can smooth temporal traffic variability and reduce service load of base stations in mobile video delivery. However, the assignment of multiple video representations to distributed servers is still a challenging question in the context of adaptive streaming, since any two representations from different videos or even from the same video will compete for the limited caching storage. It is therefore important, yet challenging, to optimally select the cached representations for each edge server in order to effectively reduce the service load of base station while maintaining a high quality of experience (QoE) for users. To address this, we study on a QoE-driven mobile edge caching placement optimization problem for dynamic adaptive video streaming that properly takes into account the different rate-distortion (R-D) characteristics of videos and the coordination among distributed edge servers. Then, by the optimal caching placement of representations for multiple videos, we maximize the aggregate average video distortion reduction of all users while minimizing the additional cost of representation downloading from the base station, subject not only to the storage capacity constraints at the edge servers, but also to the transmission and initial startup delay constraints at the users. We formulate the proposed optimization problem as an integer linear program (ILP) to provide the performance upper bound, and as a submodular maximization problem with a set of knapsack constraints to develop a practically feasible cost benefit greedy algorithm. The proposed algorithm has polynomial computational complexity and a theoretical lower bound on its performance. Simulation results further show that the proposed algorithm is able to achieve a near-optimal performance with very low time complexity. Therefore, the proposed optimization framework reveals the caching performance upper bound for general adaptive video streaming systems, while the proposed algorithm provides some design guidelines for the edge servers to select the cached representations in practice based on both the video popularity and content information.**

***Index Terms*—Mobile edge caching, adaptive video streaming, wireless video delivery, video-on-demand, submodular function maximization.**

## I. Introduction

In the last decade, mobile multimedia services, such as streaming of mobile videos, have become the main reason for the exponential growth of global mobile data traffic over cellular networks [1]. For example, as revealed by [2] in 2016, real-time entertainment that consists of streaming video and audio has become the largest traffic category on virtually every network, and its continued growth is expected to lead all the networks. Such a dramatic growth of mobile video data poses significant challenges to both the video content providers and the network service providers. One noticeable consequence is the resultant acceleration of busy-hour traffic in relation to the average traffic growth. Unlike other data traffic (e.g., web usage) that occurs throughout the day, video usage is more likely to occur during evening hours and thus has a "prime time." Globally, mobile busy-hour traffic is expected to be 88 percent higher than average-hour traffic by 2020, compared to 66 percent in 2015 [1]. Therefore, the mobile video traffic presents a high temporal variability, which incurs congestion during peak traffic hours and under-utilization during off-peak hours. To reduce the heavy traffic load of the base station and provide context-aware services in close proximity to the mobile multimedia users, mobile edge computing has been introduced to push mobile computing, network control and storage to network edges [3]. In particular, mobile edge caching (MEC) is able to utilize the storage space of edge servers across the network and to perform multimedia content placement during off-peak hours, thereby smoothing out the temporal traffic variability and reducing congestion and access latencies [4].

Simultaneously, the growing heterogeneity of user population in terms of demands for specialized video content, display devices, and access network capacity, has made the mobile video streaming a much more complex task. Adaptive streaming technique, such as the dynamic adaptive streaming over HTTP (DASH), has emerged as an effective method for video streaming over heterogeneous networks, which can improve the overall user satisfaction by offering several representations of the same video content to different clients [5]. Each representation is encoded with a pre-defined bitrate and/or resolution by the content provider. The users then select the representation that better fits to their requirements and the network conditions. Therefore, it is promising to study the potential performance gain introduced by the dynamic adaptive streaming in addition to the mobile edge caching, and to investigate the proper mobile edge caching placement schemes for dynamic adaptive streaming systems, in order to alleviate the traffic load of the base station and reduce the access latencies of the users (i.e., benefit of caching), and to satisfy heterogenous users' demands (i.e., benefit of adaptive streaming). The basic question in this context is how to place the local caches of the distributed edge servers with appropriate video representations such that the overall users' QoE in terms of video qualities and latencies is maximized, given the cache storage capacity of these edge servers. Different from the caching schemes for traditional video streaming, the number of video representations stored at the content server (which is managed by the content provider) may become extremely huge since multiple representations are stored for each video. This results in a much more difficult problem formulation with a higher computational complexity to solve it. Therefore, in adaptive streaming based MEC systems, people are not only concerned about which video should be cached at which edge server, they also want to know which representation of that video should be selected for caching.

Studies to date have investigated related work to deal with the aforementioned caching and adaptive streaming from different perspectives. For mobile video delivery, caching at distributed edge servers is demonstrated to be capable of greatly reducing the service load of base station, and replacing the usually weak backhaul connections from the base station with high-speed local links from the edge servers to guarantee the low delay requirement of users [6] [1]. An efficient caching placement strategy is designed for two-tier wireless content delivery networks to reduce the system design complexity by using separate channels for content dissemination and service [7]. For adaptive streaming, the work in [8] derives a logarithmic QoE model based on empirical results and formulates the cache management problem as a convex optimization problem. In order to cope with dynamic video segment requests, an online pre-fetching algorithm is proposed in [9] to adaptively pre-fetch adaptive streaming video segments while considering the limited bottleneck bandwidth between the content server and the edge server.

[1] According to [6], since the edge servers are much closer to the mobile users, localized high-bandwidth communication from the edge servers can be achieved through enabling high frequency reuse or high-density spatial reuse of communication resources, while the backhaul communication from the base station fails to do so.

However, the limitation of these state-of-the-art caching schemes is that the video content characteristics are not taken into account. They mainly focus on the rate (bitrate of encoded representations) and delay (transmission delay) perspectives, and thus video sources with different R-D behaviors are treated in the same way, which is not the optimal solution for the adaptive streaming scenario where different representations have different R-D behaviors.

We therefore propose in this paper to develop a novel mobile edge caching placement optimization framework for the adaptive streaming based video-on-demand (VoD) system with proper consideration of the R-D properties of the representations from different videos. Specifically, we formulate the caching placement optimization problem as an ILP, and target at maximizing aggregate average video distortion reduction of all users while taking into account the imposed constraints on the backhaul link, the edge servers' storage capacity and the users' transmission and initial startup delay. This is accomplished by the optimal assignment of adaptive streaming representations of multiple video sources to distributed edge servers. Through solving the proposed ILP to obtain the optimal solution, we are able to provide a performance upper bound for the caching placement. However, it is NP-hard and thus too time-consuming to be a practical solution for delay-sensitive video streaming. In order to reduce the execution time of the caching placement algorithm in practice, we convert the original optimization problem to an equivalent set function optimization problem and show its submodularity. By using the diminishing return property of the submodular functions, we develop a cost-benefit greedy algorithm for the caching placement, which has polynomial computational complexity and offers close-to-optimal performance (approximation ratio is theoretically proved to have a lower bound and practically shown to be above $95\%$ under different simulation settings in Section VI). We conduct extensive simulations under different system settings. The simulation results show that the proposed algorithm can scale very well with the size of the system. It also strikes the tradeoff between the algorithm execution time and the performance in terms of both the average distortion reduction per user and the base station transmission rate. Overall, the contributions of this paper can be summarized as follows.

1) Through introducing adaptive streaming to allow caching multiple representations for the same video, the proposed caching placement optimization framework addresses the users' heterogeneity issue and thus achieves an additional caching performance gain (in terms of higher average distortion reduction per user and lower base station transmission rate) over the caching schemes designed for general video files (i.e., single representation for each video). It optimally allocates the caching resources of edge servers not only among different videos, but also among multiple representations of the same video.
2) In addition to video content popularity and network conditions that are commonly considered by existing caching schemes for adaptive streaming, video content characteristics (i.e., the R-D property) are further taken into account, to assign different utilities to the representations with the same bitrate but from different videos. In this way, the actual performance of the caching system is properly evaluated in terms of the users' viewing quality.
3) To efficiently solve the proposed caching placement optimization, we convert it to an equivalent submodular maximization problem with a set of knapsack constraints. We develop a polynomial-time greedy algorithm and provide a theoretical proof on the lower bound of its approximation ratio.

The rest of this paper is organized as follows. Section II reviews the related works in literature. In Section III, we introduce the mobile edge caching placement framework and related system models. In Section IV, we formulate the caching problem as an ILP by considering the users' QoE and edge servers' cache capacity constraints. In Section V, we transform the original ILP to an equivalent submodular maximization problem, and develop a practical approximation algorithm to solve this problem with close-to-optimal performance. Section VI presents experimental results, and evaluates the gains of the proposed algorithm compared to existing algorithms. The concluding remarks are given in Section VII.

## II. Related Work

The idea of using mobile edge caching to support the cellular level communication has been recently explored in [6], [10]–[18]. In [10], Liu *et al.* summarize the design aspects and challenges of mobile edge caching. They further reveal that caching at the wireless edge for 5G cellular networks is still an open problem since the unique limitations in wireless networks due to the architecture and channel (such as the network topology, link interference, users' mobility, and limited battery) must be considered when designing an appropriate caching placement strategy. In [11], the authors study a caching scheme for the 5G edge cloud network where contents are stored with a price determined by the mobile network operator. The novel *FemtoCaching* architecture in [6], [12] proposes mobile edge caching at the small-cell access points, by compensating the backhaul capacity with the storage capacity at the mobile edge to efficiently handle some highly predictable bulky traffic (e.g., VoD traffic). The mobile video caching placement over distributed edge servers is essentially used to minimize the average downloading delay of users. The authors in [13] develop a distributed caching optimization algorithm via belief propagation for the heterogeneous cellular networks with edge servers, in order to minimize the overall downloading delay. Sengupta *et al.* [14] study the fundamental information theoretic limit of mobile edge caching, revealing the optimal tradeoff between the latencies and cache sizes. The work in [15] formulates a joint routing and caching problem that targets at maximizing the fraction of content requests served locally by the deployed edge servers, under the consideration of some important features such as the storage and bandwidth capacities of edge servers, and the content request patterns of users. By further incorporating the users' link interference issue, a joint caching, routing and channel assignment problem is proposed in [16] to maximize the throughput of the video delivery over coordinated small-cell cellular systems.While most of the above works assume a priori knowledge about the content popularity, the authors in [17], [18] propose a context/trend-aware caching scheme to predict the popularity information based on the users' context (e.g., his/her personal characteristics, equipment, or external factors), which explicitly learns the context-specific popularity of video content through online learning and uses it to determine the caching replacement decision. The online learning here indicates that the context information becomes available in a sequential order and is used to update the best predictor for the short-term popularity of content at each time step, as opposed to the learning techniques that generate the best predictor by learning on the entire training set at only one dedicated training phase. However, all these above studies only focus on the caching assignment problems for general (video) files. This is however not sufficient in the context of adaptive video streaming [10], where appropriate bitrate representations need to be carefully determined and pre-fetched in the edge servers.

In another line of research, some works have been done to leverage caching in the dynamic adaptive video streaming system [8], [9], [19]–[25]. From the rate adaptation perspective, Lee *et al.* [19] investigate the bitrate oscillation and sudden rate change problem occurring through the interaction between the clients and caches, and

TABLE I
COMPARISON WITH THE MOST RELEVANT WORKS ON MOBILE EDGE CACHING FOR VIDEO STREAMING.

| | This work | [6], [12] | [13], [15], [16] | [20] | [9] | [25] | [23], [24] |
|---|---|---|---|---|---|---|---|
| Applicable to adaptive streaming | Yes | No | No | Yes | Yes | Yes | Yes |
| Optimal performance upper bound | Yes | No | Yes | Yes | Yes | No | No |
| Approximation algorithm guarantee | Yes | Yes | Yes | N/A | Yes | No | No |
| Operational-cost/rate-cost aware | Yes | Yes | Yes | Yes | Yes | No | Yes |
| Video content characteristics aware | Yes | No | No | No | No | Yes | No |

propose an approach that uses shaping to eliminate such oscillations. Jin *et al.* [20] apply caching to adaptive streaming, and study the optimal transcoding and caching allocation scheme in media cloud in order to minimize the total operational cost of delivering on-demand adaptive video streaming, with the assumption that each mobile user accesses one edge server for video downloading. Gao *et al.* [21] investigate the tradeoff between storage and transcoding computation in the cloud, and propose a cost-efficient partial transcoding scheme for content management based on user viewing patterns. Zhao *et al.* [22] further develop a video segment-based caching strategy for multiple representation VoD systems to minimize the storage and transcoding costs. In order to cope with dynamic requests, the work in [9] proposes an online pre-fetching algorithm to adaptively pre-fetch adaptive streaming video segments while respecting the limited bottleneck bandwidth between the content server and the edge server. To improve the users' QoE, the authors in [8] derive a logarithmic QoE model based on empirical results and formulate a cache management problem for adaptive streaming as a convex optimization problem, thereby providing an analytical framework for this engineering problem. The work in [23] proposes an in-network video caching policy for information centric networks to enhance users' QoE in terms of average user throughput, based on the content popularity distribution. A QoE-driven DASH video caching and adaptation algorithm is proposed in [24] to make the caching and replacement decision based on the content context (e.g., segment popularity) and the network context (e.g., downlink bandwidth). However, all these works only focus on the operational-cost/rate perspective and thus neglect the video content characteristics of the representations from different video contents. Here, the video content means the distinct foreground, background and motion in the video, which results in different rate-distortion (R-D) behaviors (considered as the video content characteristics) for different video sources after encoding. In other words, this difference of video content (or R-D behaviors) between different videos is not considered in the above works, where the multiple representations encoded from different raw videos but with the same bitrate are assumed to have the same system utility. Therefore, their caching performance depends only on the video content popularity and network conditions. However, as will be justified by the experimental results in Section VI, it is only by carefully considering the video content characteristics (i.e., the R-D behavior) that the actual performance of the caching system can be properly evaluated in terms of user utility.

In our previous work [25], we have partially addressed this issue by proposing a wireless video caching placement optimization problem for dynamic adaptive video streaming and a fast approximation algorithm to minimize the average video distortion of all clients, under the edge servers' storage capacity constraints. In this work, we further provide a general optimization formulation as an ILP along with its optimal solution as a performance upper bound. In addition, we also take into account other QoE metrics, such as the initial startup delay, in order to better reflect the actual utility of each video stream. Finally, we study in detail the approximation algorithm for the cache allocation, and provide a theoretical lower bound on its performance.

In summary, Table I lists the differences between this work and the most relevant papers in the literature on mobile edge caching for video streaming. Within these references, [6] and [12] are the most related model. Through the comparison in Table I, it can be seen that the work in [6] and [12] is a caching scheme designed for general video files (i.e., single representation for each video) and only considers video content popularity distribution and network conditions, while this work addresses the caching resource allocation among different videos and different representations of adaptive streaming through the consideration of video content characteristics (i.e., the R-D property). In addition, the femto-cache algorithm proposed in [6] and [12] has been selected as a comparison algorithm in Section VI, which justifies that compared to the femto-cache algorithm, this work can achieve a higher caching performance gain in terms of higher average distortion reduction per user and lower base station transmission rate.

## III. FRAMEWORK AND SYSTEM MODELS

In this section, we introduce the mobile edge caching placement framework for dynamic adaptive video streaming systems and related models.

### A. Framework

Consider a wireless adaptive streaming based VoD system as illustrated in Fig. 1. Suppose that the base station stores $F$ video files, each of which is encoded into $M$ different representations. $S$ edge servers with certain capabilities of pre-fetching video content are deterministically placed in the wireless coverage region of the base station, and are assumed to connect to the base station through single hop transmission. If the connection between the base station and edge servers in some cases is multi-hop, the multi-hop connection characteristics can be considered as the end-to-end transmission rate between them. These edge servers are geographically closer to the mobile users and enable high-density spatial reuse of the wireless resources with high-speed localized communication, which is usually assumed to be much faster than the backhaul links connected to the base station [12]. For the VoD service with a priori knowledge of the video popularity distribution, some popular video files can be pre-fetched by the edge servers during the off-peak hours to relieve the service load of the base station and to replace the weak backhaul communication.

The mobile edge caching placement criteria for adaptive streaming are as follows. Whenever a mobile user sends a playback request for a specific video, it attempts to download the highest possible quality representation from its adjacent edge servers in accordance with the content placement and the available download link capacity. If the same high quality representation is cached in multiple edge servers, the user might want to download it from the edge server with the highest transmission rate, in order to reduce the initial startup delay. That is, the user will first determine whether there is a representation with the highest bitrate available at one of its adjacent edge servers and the download of this representation can be supported by the link capacity with an acceptable downloading delay. If yes, the user could download and playback that representation; otherwise, it would make a further selection for the representation with the next lower

bitrate. This determination will continue until a representation with an affordable bitrate is found at an edge server or the representation with the smallest bitrate is reached. When no representation of the requested video is available at any adjacent edge server, the user has to turn to the base station and download the representation with the highest bitrate that could be afforded by the backhaul link connected to the base station. However, downloading from the base station will result in a much more expensive transmission cost since the backhaul communication resource is typically very limited compared to the high-speed links offered by the adjacent edge servers.

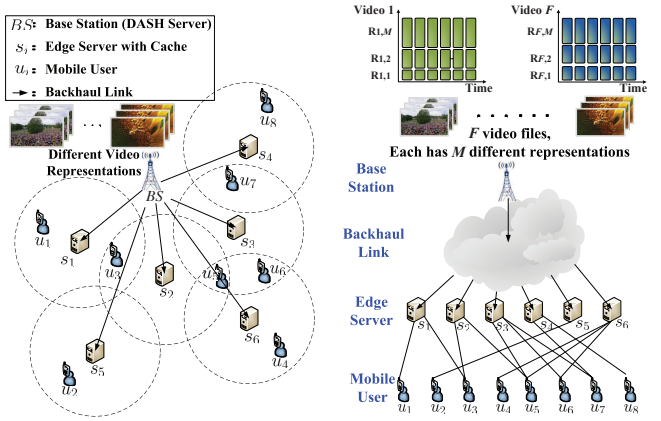
### B. System Models

We now describe in more detail the model that we consider in this work, and introduce the notation.

Let first $\mathcal{F}$ denote the set of $F$ video files that are offered to the users. Any video file $f \in \mathcal{F}$ is encoded into a set of $M$ representations $\mathcal{Z}_f = \{z_{f,m} | \forall m = 1, 2, \ldots, M\}$ with the $m$-th representation $z_{f,m}$ having an encoding bitrate being $R_{f,m}$. We further suppose that this set is sorted in a decreasing order of the encoding bitrate, i.e., $R_{f,i} > R_{f,j}, \forall 1 \leq i < j \leq M$. Therefore, the complete set including all representations for all the video files can be denoted as $\mathcal{Z} = \cup_{f \in \mathcal{F}} \mathcal{Z}_f$. For the sake of simplicity, and without loss of fundamental generality, we adopt the assumption from [20], that each video file has the same length $T$. Such assumption is mainly proposed for the notational convenience, and could be easily lifted by breaking a longer file into multiple files of the same length [12]. If in some scenarios the video lengths are significantly heterogeneous and this assumption becomes no longer reasonable, we can use the notation $T_f$ to represent the length of video file $f$ in the cache capacity constraint of **ILP** in Eq. (8b) (or its equivalent submodular problem in Eq. (12b)), which would not fundamentally change the corresponding analysis and algorithm design.

To illustrate the connection between the edge servers and the users, the wireless network is defined by a bipartite graph $\mathcal{G}_{su} = (\mathcal{S}, \mathcal{U}, \mathcal{E}_{su})$, where $\mathcal{S}$ represents the set of $S$ edge servers, $\mathcal{U}$ denotes the set of $U$ mobile users, and a graph edge $(s, u) \in \mathcal{E}_{su}$ indicates that a wireless communication link exists from the edge server $s \in \mathcal{S}$ to the user $u \in \mathcal{U}$. The download link transmission rate of the wireless link $(s, u)$ is denoted by $c_{(s,u)}$[2]. For each edge server $s \in \mathcal{S}$, the cache storage capability is constrained by the capacity $B_s$. Finally, we denote by $\mathcal{N}(u)$ the neighboring edge servers of user $u \in \mathcal{U}$. We assume that $\mathcal{N}(u)$ is sorted in a decreasing order of the download link capacity, such that $(i)_u \in \mathcal{N}(u)$ represents the edge server with the $i$-th largest capacity of the link to the user $u$. In this paper, we study the caching system with the caching placement decision to be made for a certain time period (e.g., several hours during the peak hours, or even several days), during which the average demand for the set of $F$ video files is assumed to be known in advance, as in [12], [20], [29]. In this way, the backhaul is only used to refresh the caches at the rate at which the user request distribution evolves over time, which is a much slower process than the time scale at which the users place their requests [12]. Therefore, we adopt the assumption from [12], [20], that users' requests are statistically independent and a probability mass function $P_{u,f}$ is used to represent the average probability that the video file $f \in \mathcal{F}$ is requested by the user $u \in \mathcal{U}$ within this time period. This independent user request model is an acceptable approximation in an average sense or when the content popularity variation over time is relatively slow.

[2]In this paper, we assume that we have detected and known the accurate channel state information (CSI) for the upcoming transmission frame and that the transmission rate $c_{(s,u)}$ is known a priori. For the time-vary wireless channel when $c_{(s,u)}$ is not perfectly known and may change over time, channel prediction techniques [26] can be used to estimate the link transmission rate. For example, the finite state Markov channel model [27], [28] is widely adopted as a good approximation in modeling and predicting the time-varying processes of wireless links. However, the detailed description of these channel prediction techniques is beyond the scope of this paper.




(a) (b)

Fig. 1. (a) Example of the system layout, where mobile users are randomly distributed, while edge servers are connected to the base station with backhaul links and can be deterministically placed in the coverage region. (b) The connectivity bipartite graph indicating how mobile users are connected to the edge servers.

We further consider a caching system where a representation of a video file is either cached fully (i.e., the whole representation of the length $T$) or not cached at all in any edge server[3], the representation placement strategy can be represented by a bipartite graph $\mathcal{G}_{z_{f,m},s} = (\mathcal{Z}, \mathcal{S}, \mathcal{E}_{z_{f,m},s})$ between vertices representing edge servers in $\mathcal{S}$, and vertices describing video representations in $\mathcal{Z}$. An edge $(z_{f,m}, s) \in \mathcal{E}_{z_{f,m},s}$ is drawn when $z_{f,m}$ (i.e., the $m$-th representation of video file $f$) is stored in the cache of edge server $s$. To better understand the representation placement strategy as shown by the bipartite graph, we can further denote $\boldsymbol{A}_{F \times M \times S}$ as the $F \times M \times S$ adjacency matrix of $\mathcal{G}_{z_{f,m},s}$, such that $\forall s \in \mathcal{S}$, $a^s_{f,m} = 1$ indicates that an edge $(z_{f,m}, s) \in \mathcal{E}_{z_{f,m},s}$ exists and $a^s_{f,m} = 0$ denotes the absence of an edge between $z_{f,m}$ and $s$, i.e.,

$$a^s_{f,m} = \begin{cases} 1, & \text{if the edge server } s \text{ caches the } m\text{-th} \\ & \text{representation of video } f; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

### C. Quality-of-Experience Models

According to [32], both the initial startup delay (the waiting time interval between the client's request and the beginning of the playback) and the average video quality (the average video distortion) are the key factors that affect the quality of experience (QoE) of video streaming services.

For each user $u \in \mathcal{U}$, the initial startup delay constraint requires that the waiting time interval between submitting a request and the actual video playback should not exceed the maximum tolerable

[3]In some scenarios where the sizes of video files are very large (e.g., HD videos, or video length $T$ is too long) and the caching storage resource becomes the critical concern, we can alternatively adopt the partial caching strategy that caches the first portion of the same length $T'$ ($T' \ll T$) for each representation of each video. The reason is as follows. Based on the studies on users' behavior and viewing patterns in some practical VoD systems, such as YouTube [30] and PPTV [31], it is observed that usually users only watch a small portion of the full content of a video. For example, statistics in [30] show that 95% of the views last shorter than 200 seconds. Therefore, the consumption of caching storage greatly decreases by only partially caching the first $T'$ seconds of each representation (e.g., $T' = 200$ s), and the system is still efficient since most of the time (e.g., $> 95\%$) the users are satisfied with the partially cached content.

waiting time of that user, which is denoted as $d_{u,\max}$. Let us assume first that the video representation $z_{f,m} \in \mathcal{Z}$ is available in the cache of user $u$'s adjacent edge server $s \in \mathcal{S}$. Let us further denote with $\Delta T$ the time fraction within a video file that is required to be buffered by the user before the actual playback starts on the user's screen. Then the initial startup delay experienced by the user $u$ to download the representation $z_{f,m}$ from the edge server $s$ is:

$$d^s_{u,f,m} = \frac{R_{f,m} \cdot \Delta T}{c(s,u)}, \forall u \in \mathcal{U}, \forall z_{f,m} \in \mathcal{Z}, \forall s \in \mathcal{S}. \quad (2)$$

Here, we set the transmission rate of links from the non-adjacent edge servers of a user to a small positive value that is arbitrarily close to zero, i.e., for all $s \notin \mathcal{N}(u)$ we have $c_{(s,u)} = \varepsilon$, where $\varepsilon \to 0$ and accordingly $d^s_{u,f,m} \to +\infty$. Similarly, when the requested video is not available in the edge servers, the initial startup delay experienced by the user $u$ to download $z_{f,m}$ from the base station is:

$$d_{u,f,m} = \frac{R_{f,m} \cdot \Delta T}{c(BS,u)}, \forall u \in \mathcal{U}, \forall z_{f,m} \in \mathcal{Z}, \quad (3)$$

where $c(BS,u)$ is the download link transmission rate of the wireless link connecting the base station and the user.

Then, we use a general rate-distortion function $D_{\max} - \Delta D_f(R_{f,m})$ to denote the distortion of the $m$-th representation of the video $f$ with the encoding bitrate $R_{f,m}$, where $D_{max}$ and $\Delta D_f(R_{f,m})$ represent a constant maximal distortion when no video is decoded and the distortion reduction (or quality improvement) after successfully decoding this representation, respectively.By utilizing the R-D model in [33], $\Delta D_f(R_{f,m})$ can be expressed as:

$$\Delta D_f(R_{f,m}) = D_{\max} - D_0 - \frac{\theta}{R_{f,m} - R_0} \quad (4)$$

where the variables, $\theta$, $R_0$ and $D_0$, are empirical parameters that depend on the actual video content; they can be estimated as the fitting parameters from the empirical rate-distortion curves of different videos by using regression techniques.

## IV. QoE-Driven Caching Placement Optimization Problem

In this section, we describe the QoE-driven mobile edge caching placement optimization problem for adaptive streaming, and formulate it as an ILP.

### A. Problem Description and Challenges

The QoE-driven mobile edge caching placement problem for adaptive streaming can be summarized as follows: given the representation set of source video files, the file popularity distribution, the edge server storage capacity and the network topology, how to place the representations of the video files in the distributed edge servers such that the total system utility (which is defined by Eqs. (7) and (8a) in the next subsection) is maximized subject to the caching capacity constraint of each edge server and the downloading delay requirement of each user.

If each video file has only one representation and each user has only access to one edge server, the optimal placement strategy becomes simple and straightforward. That is, each edge server should cache as many of the most popular video files as possible until its storage is full. However, for the case of dense edge server deployment where each user can have access to more than one edge servers, the optimal content placement strategy becomes highly nontrivial. Furthermore, if each video file is available in different representations with different bitrates, the optimal placement problem becomes even more complicated.

Compared to the caching problem with general files, the fundamental technical challenges introduced by the adaptive video streaming, i.e., multiple representations of a video file need to be cached, can be explained as follows. The general file caching problem usually addresses the caching resource competition issue among different files by placing appropriate files in the distributed edge servers. It is also based on the assumption that there is no difference between different files in terms of the system utility, i.e., downloading a different file would lead to the same utility improvement (e.g., the increase of hit ratio). When the adaptive video streaming is taken into account, however, people are not only concerned with which video file should be cached at which edge server, they also want to know which representation(s) should be selected to cache in order to maximize the overall system utility. This means that not only different video files, but also the multiple representations of the same video file will compete for the caching resource at the edge servers. In addition, due to the difference of video content characteristics, downloading the same bitrate representation of different video files would also result in different utility improvement (e.g., the distortion reduction). Even for the same video file, the caching resource allocation problem becomes more complicated since the relationship between the utility improvement (e.g., the distortion reduction) and the bitrate of the different representations is nonlinear and presents the diminishing return property. It should be noted that all of the above issues introduced by the adaptive streaming cannot be straightforwardly addressed by the general file caching problem, which motivates us to study the following caching placement optimization problem for adaptive streaming.

### B. System Utility Function

First, we introduce two sets of auxiliary binary variables:

$$\beta^s_{u,f,m} = \begin{cases} 1, & \text{if user } u \text{ gets the } m\text{-th representation} \\ & \text{of video } f \text{ from edge server } s; \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$$\gamma_{u,f,m} = \begin{cases} 1, & \text{if user } u \text{ gets the } m\text{-th representation} \\ & \text{of video } f \text{ from the base station;} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

We then define the following utility function, based on both the average video distortion reduction experienced by the user $u$ and the cost of the representation downloading either from the edge server or the base station:

$$Q_u = \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \sum_{s\in\mathcal{N}(u)} \beta^s_{u,f,m} \cdot P_{u,f} \cdot [\Delta D_f(R_{f,m}) - \eta_0 \cdot R_{f,m}]$$
$$+ \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \gamma_{u,f,m} \cdot P_{u,f} \cdot [\Delta D_f(R_{f,m}) - \eta \cdot R_{f,m}] \quad (7a)$$
$$\approx \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \sum_{s\in\mathcal{N}(u)} \beta^s_{u,f,m} \cdot P_{u,f} \cdot \Delta D_f(R_{f,m})$$
$$+ \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \gamma_{u,f,m} \cdot P_{u,f} \cdot [\Delta D_f(R_{f,m}) - \eta \cdot R_{f,m}]. \quad (7b)$$

As usually done in many rate-distortion optimization problems [34], in the utility function defined in Eq. (7a), we impose the bandwidth constraints (from the edge servers and the BS) as the cost penalty, rather than putting them as hard constraints. It represents a typical optimization objective that trades bandwidth (resource cost) for video quality. Specifically, $[\Delta D_f(R_{f,m}) - \eta_0 \cdot R_{f,m}]$ in the first term of Eq. (7a) includes the video distortion reduction $\Delta D_f(R_{f,m})$

of downloading the representation $z_{f,m}$, and a transmission cost penalty $\eta_0 \cdot R_{f,m}$ where $\eta_0$ is the unit price parameter corresponding to the representation downloading of $z_{f,m}$ from the adjacent edge servers. As constrained by Eq. (8f), for any user $u \in \mathcal{U}$ and any video file $f \in \mathcal{F}$, at most one $\beta^s_{u,f,m}, \forall m = 1, 2, \ldots, M, \forall s \in \mathcal{N}(u)$ equals to 1. Therefore, the weighted summation (where the weight is the video request probability $P_{u,f}$) over all $F$ video files, $\sum_{f\in\mathcal{F}}\sum_{m=1}^{M}\sum_{s\in\mathcal{N}(u)}\beta^s_{u,f,m}\cdot P_{u,f}\cdot[\Delta D_f(R_{f,m}) - \eta_0 \cdot R_{f,m}]$, represents the average video distortion reduction plus the average transmission cost penalty experienced by user $u$ downloading requested video representations from its adjacent edge servers. Likewise, the second term in Eq. (7a) represents the average video distortion reduction plus the average transmission cost penalty experienced by user $u$ downloading requested video representations from the base station. Due to the limited bandwidth available in the backhaul channel, the unit price for downloading from the base station is much higher than the unit price for accessing the adjacent edge servers (i.e., $\eta \gg \eta_0$)[4]. As a consequence, the overall caching system will prefer to store representations in the edge servers, since downloading the same representation from an edge server achieves the same distortion reduction gain while the transmission cost is much lower. Users will only access the base station for representation downloading in some rare cases when they are highly rewarded. This happens either when there is no representation of the requested video cached in their adjacent edge servers, or when the cached content has a very poor quality and the distortion reduction gain of a better quality representation is so high that downloading it from the base station with a higher transmission cost is worthy for the overall utility improvement. For the sake of simplicity, hereinafter, we assume that $\eta_0 \to 0$ and $\eta$ is a positive constant, and thus define the utility function as shown in Eq. (7b).

### C. Optimization Problem Formulation

Mathematically, the QoE-driven mobile edge caching placement problem for adaptive streaming can be formulated as an integer linear program (ILP), as follows:

$$\textbf{ILP:} \quad \max_{\boldsymbol{A},\boldsymbol{\beta},\boldsymbol{\gamma}} \quad \sum_{u\in\mathcal{U}} Q_u \tag{8a}$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}}\sum_{m=1}^{M} a^s_{f,m}\cdot R_{f,m}\cdot T \le B_s, \forall s\in\mathcal{S}, \tag{8b}$$

$$\beta^s_{u,f,m}\cdot d^s_{u,f,m} \le d_{u,\max}, \forall u\in\mathcal{U}, \forall z_{f,m}\in\mathcal{Z}, \forall s\in\mathcal{S}, \tag{8c}$$

$$\gamma_{u,f,m}\cdot d_{u,f,m} \le d_{u,\max}, \forall u\in\mathcal{U}, \forall z_{f,m}\in\mathcal{Z}, \tag{8d}$$

$$\beta^s_{u,f,m} \le a^s_{f,m}, \forall u\in\mathcal{U}, \forall z_{f,m}\in\mathcal{Z}, \forall s\in\mathcal{S}, \tag{8e}$$

$$\sum_{m=1}^{M}\gamma_{u,f,m} + \sum_{m=1}^{M}\sum_{s\in\mathcal{N}(u)}\beta^s_{u,f,m} \le 1, \forall u\in\mathcal{U}, \forall f\in\mathcal{F}, \tag{8f}$$

$$\beta^s_{u,f,m} \in \{0,1\}, \forall u\in\mathcal{U}, \forall z_{f,m}\in\mathcal{Z}, \forall s\in\mathcal{S}, \tag{8g}$$

$$\gamma_{u,f,m} \in \{0,1\}, \forall u\in\mathcal{U}, \forall z_{f,m}\in\mathcal{Z}, \tag{8h}$$

$$a^s_{f,m} \in \{0,1\}, \forall z_{f,m}\in\mathcal{Z}, \forall s\in\mathcal{S}. \tag{8i}$$

[4]For the sake of simplicity, we assume in this paper that the unit downloading price $\eta_0$ is the same for different edge servers, since the downloading cost of the same representation from different edge servers differs very slightly compared to the much larger downloading cost from the base station. This assumption could be lifted by assigning a different unit downloading price $\eta^s_0$ in Eq. (7a) to an edge server $s$. Then, the ILP in Eq. (8) can be similarly solved by setting the optimization objective according to Eq. (7a). For the equivalent submodular maximization problem and its approximation algorithm, we only need to re-sort the set of neighboring edge servers $\mathcal{N}_u$ for each user $u$, in such a way that $(i)_u \in \mathcal{N}(u)$ represents the edge server offering the $i$-th smallest unit downloading price.

In the above **ILP**, the objective is to maximize the aggregate utility defined in Eq. (7b), or equiavelent to maximize the average video distortion reduction of all users (which is equivalent to minimizing the aggregate average video distortion) while minimizing the transmission cost of the representation downloading from the base station. The decision variables are the representation placement strategy represented by the adjacency matrix $\boldsymbol{A}_{F\times M\times S} \in \{0,1\}^{F\times M\times S}$ and the sets of auxiliary binary variables $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. The constraint in Eq. (8b) represents the cache capacity constraints of each edge server, where $T$ is the time duration of each video file. The startup delay constraints in Eqs. (8c) and (8d) specify that the initial startup delay experienced by the user $u$ to download the representation $z_{f,m}$ either from the edge server $s$ or the base station should not exceed the maximum tolerant waiting time $d_{u,\max}$. The constraint in Eq. (8e) sets up a consistent relationship between the decision matrix $\boldsymbol{A}$ and auxiliary variables $\boldsymbol{\beta}$, ensuring that the representation selected by a user is already cached and available at the edge server $s$. The constraint in Eq. (8f) imposes that for any video $f$, the user $u$ can only download at most one representation from at most one edge server (or the base station), to avoid duplicated downloading of multiple representations for the same video or the same representation from multiple edge servers (or the base station). Together with the startup delay constraints in Eqs. (8c) and (8d), it ensures that only one representation will be downloaded by the user $u$ for the video $f$. Furthermore, this representation is the largest possible bitrate representation under the user's download link capacity and the startup delay constraints, since otherwise the value of the objective function in Eq. (8a) decreases, which indicates a non-optimal solution. The constraints in Eqs. (8g)-(8i) define the binary decision and auxiliary variables, respectively.

The optimal solution of the **ILP** can be obtained by the generic solver IBM ILOG CPLEX [35], using a branch-and-cut search. The branch-and-cut procedure follows a search tree consisting of nodes, each of which represents a relaxed LP subproblem to be solved. It then involves running a branch and bound algorithm to create two new nodes from a parent node, and adding additional cutting planes to tighten the LP relaxations and reduce the number of branches required to solve the original ILP. In general, the branch-and-cut search requires exponential computational complexity to achieve the optimal solution in the worst case [36], [37]. Therefore, the **ILP** problem in Eq. (8) is NP-hard. Specifically, it can be observed that the cardinality of the decision variables $\boldsymbol{A}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ is $FMS$, $UFMS$, and $UFM$, respectively. By using the branch and bound method for the binary decision variables, in the worst case, the number of nodes observed by the CPLEX solver would be upper bounded by $2^{FMS}\times 2^{UFMS}\times 2^{UFM}$. At each node the solver needs to solve a relaxed LP problem with the SIMPLEX method. This corresponds to an exponential computational complexity $O(2^{2U\cdot 3F\cdot 3M\cdot 2S})$ and thus incurs an incredibly long execution time when the problem scale becomes large.

## V. Equivalent Submodular Maximization Problem and Algorithm Design

In order to efficiently cope with the difficulties of solving the **ILP** in Eq. (8), in this section, we convert it to an equivalent set function optimization problem. We prove that it is a submodular maximization problem over independence constraints. We finally develop new practically efficient algorithms with polynomial computational time complexity and theoretical approximation guarantees.

### A. Equivalent Problem Formulation as a Set Function Optimization

In accordance with the adjacency matrix $\boldsymbol{A}_{F\times M\times S}$ in the **ILP** in Eq. (8), the finite ground set of the equivalent set function

optimization problem can be viewed as:

$$\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_s, \ldots \mathcal{V}_S\}, \tag{9}$$
$$\mathcal{V}_s = \{v^s_{1,1}, \ldots, v^s_{1,M}, \ldots, v^s_{f,m}, \ldots, v^s_{F,1}, \ldots, v^s_{F,M}\}, \forall s \in \mathcal{S},$$

where the ground set is partitioned into $S$ disjoint subsets. Each subset $\mathcal{V}_s$ denotes the full set of all representations of all files that may be cached on the edge server $s$, and the element $v^s_{f,m}$ represents the placement of the $m$-th representation of video file $f$ (i.e., $z_{f,m}$) on the cache of the edge server $s$. For a given adjacency matrix $\boldsymbol{A}_{F\times M\times S}$, the corresponding representation placement set $\mathcal{A} \subseteq \mathcal{V}$ can be defined in such a way that $v^s_{f,m} \in \mathcal{A}$ corresponds to the case $a^s_{f,m} = 1$ and vice versa.

When initial startup delay constraints are taken into account, the feasible set should be re-defined by eliminating the elements that violate the maximum tolerance of the initial startup delay from the ground set $\mathcal{V}$ in Eq. (9). From the perspective of users, for any $u \in \mathcal{U}$, the initial startup delay constraint indicates that a representation that could be downloaded from an edge sever within the maximum delay bound is considered feasible and might contribute to the aggregate expected distortion reduction. In the **ILP** in Eq. (8), such a constraint is indicated by Eq. (8c), which corresponds to a feasible subset of the ground set $\mathcal{V}$:

$$\Omega_u = \left\{ v^s_{f,m} \in \mathcal{V} \middle| d^s_{u,f,m} \le d_{u,\max}, \forall s \in \mathcal{S}, \forall z_{f,m} \in \mathcal{Z} \right\} \subseteq \mathcal{V}, \quad \forall u \in \mathcal{U}. \tag{10}$$

It should be noted that for a given representation set $\mathcal{F}_\mathcal{M}$ and known transmission rate for links between $\mathcal{S}$ and $\mathcal{U}$, the feasible subset $\Omega_u$ is also given with respect to the value of $d_{u,\max}$. Accordingly, the utility function of user $u$ in Eq. (7) can be rewritten in terms of the set function, by also considering the initial startup delay constraints, as:

$$\begin{aligned} Q_u(\mathcal{A}) = & \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{N}(u)|} \left[ \prod_{n=1}^{m-1} \prod_{j=1}^{|\mathcal{N}(u)|} (1 - \mathbf{1}|_{v^{(j)_u}_{f,n} \in (\mathcal{A}\cap\Omega_u)}) \right] \\ & \cdot \left[ \prod_{j=1}^{i-1} (1 - \mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)}) \right] \cdot \mathbf{1}|_{v^{(i)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)} \cdot P_{u,f} \cdot \Delta D_f(R_{f,m}) \\ & + \sum_{f\in\mathcal{F}} \left[ \prod_{m=1}^{M} \prod_{j=1}^{|\mathcal{N}(u)|} (1 - \mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)}) \right] \\ & \cdot P_{u,f} \cdot [\Delta D_f(R_{f,m^*}) - \eta \cdot R_{f,m^*}]. \end{aligned} \tag{11}$$

The definition of Eq. (11) follows the distributed caching placement criterion in Section III-A. In Eq. (11), $\mathbf{1}|_{x\in\mathcal{X}}$ is an indicator function, which is 1 if $x \in \mathcal{X}$ and 0 otherwise; and the term $[\prod_{n=1}^{m-1} \prod_{j=1}^{|\mathcal{N}(u)|} (1-\mathbf{1}|_{v^{(j)_u}_{f,n} \in (\mathcal{A}\cap\Omega_u)})] \cdot [\prod_{j=1}^{i-1} (1-\mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)})] \cdot \mathbf{1}|_{v^{(i)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)} = 1$ is the indicator function defined over the feasible placement set $\mathcal{A}\cap\Omega_u$ for the case where the $m$-th representation of video file $f$ is the best representation that user $u$ could find in its neighboring edge servers while the initial startup delay constraint is satisfied, and this representation is at the cache of edge server $(i)_u$. In particular, $[\prod_{n=1}^{m-1} \prod_{j=1}^{|\mathcal{N}(u)|} (1 - \mathbf{1}|_{v^{(j)_u}_{f,n} \in (\mathcal{A}\cap\Omega_u)})] = 1$ indicates that no representation with an index smaller than $m$ is available at any of the adjacent edge servers; and $[\prod_{j=1}^{i-1} (1 - \mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)})] = 1$ indicates that the $m$-th representation is not available at any of the edge servers with a larger download link rate (shorter initial startup delay) than the edge server $(i)_u$. The term $[\prod_{m=1}^{M} \prod_{j=1}^{|\mathcal{N}(u)|} (1 - \mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)})] = 1$ indicates that no representation of video file $f$ can be found in any neighboring edge server of user $u$, and the user $u$ will download from the base station the representation $z_{f,m^*}$ that has the highest bitrate while still respecting the initial startup delay constraint, namely $z_{f,m^*} = \arg\max_{\{z_{f,m}\in\mathcal{Z},\ d_{u,f,m}\le d_{u,\max}\}} R_{f,m}$.

Therefore, the original optimization problem **ILP** in Eq. (8) can be reformulated as a constrained set function optimization problem that leads to the same solution of the **ILP** based on the distributed caching placement criterion in Section III-A, as follows:

$$\textbf{SUB:} \quad \max_{\mathcal{A}\subseteq\mathcal{V}} Q(\mathcal{A}) = \sum_{u\in\mathcal{U}} Q_u(\mathcal{A}) \tag{12a}$$

$$\text{s.t.} \quad \mathcal{A} \in \mathcal{I}, \tag{12b}$$

$$\mathcal{I} = \left\{ \mathcal{A}' \subseteq \mathcal{V} \middle| \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \mathbf{1}|_{v^s_{f,m}\in\mathcal{A}'} \cdot R_{f,m} \cdot T \le B_s, \forall s \in \mathcal{S} \right\}.$$

Comparing the original problem **ILP** in Eq. (8) with the equivalent set function optimization formulation **SUB** in Eq. (12), it can be seen that the objective function and the first constraint in the problem **ILP** in Eq. (8) are transformed to Eqs. (12a) and (12b) in problem **SUB**, respectively. The initial startup delay constraint of each user $u$ in Eq. (8c) is preserved by the feasible subset $\Omega_u$ applied in the objective function $Q_u(\mathcal{A})$ as defined in Eq. (11), while the delay constraint in Eq. (8d) is ensured by the definition of $z_{f,m^*}$ in Eq. (11). The constraints in Eqs. (8e) and (8f) are also guaranteed since $Q_u(\mathcal{A})$ in Eq. (11) is derived according to the distributed caching placement criterion in Section III-A.That is, for each video, only one achievable representation with the highest bitrate will be selected for each user with its coefficient, either $[\prod_{n=1}^{m-1} \prod_{j=1}^{|\mathcal{N}(u)|} (1 - \mathbf{1}|_{v^{(j)_u}_{f,n} \in (\mathcal{A}\cap\Omega_u)})] \cdot [\prod_{j=1}^{i-1} (1-\mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)})] \cdot \mathbf{1}|_{v^{(i)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)}$ or $[\prod_{m=1}^{M} \prod_{j=1}^{|\mathcal{N}(u)|} (1-\mathbf{1}|_{v^{(j)_u}_{f,m} \in (\mathcal{A}\cap\Omega_u)})]$, in Eq. (11) being one, while the coefficients of the other representations are all zeros.

### *B. Submodular Maximization Problem*

Submodularity, often viewed as a discrete analogue of convexity, plays a central role in discrete optimization. Its characterizing property, diminishing marginal returns, makes submodular maximization an efficient approach for many real-world applications, including approximation algorithms and many challenging problems in machine learning. We show now that problem **SUB** in Eq. (12) is a submodular maximization problem. We first review and include the definition of submodular functions according to [38]–[40].

**Definition 1.** *Submodularity: Let $\mathcal{V}$ be a finite ground set, and a set function $g : 2^\mathcal{V} \to \mathbb{R}$ is submodular if and only if for any sets $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{V}$ and for any element $v \in (\mathcal{Y} \setminus \mathcal{X})$, we have*

$$g(\mathcal{X}) + g(\mathcal{Y}) \ge g(\mathcal{X}\cup\mathcal{Y}) + g(\mathcal{X}\cap\mathcal{Y}), \tag{13}$$

*or equivalently*

$$g(\mathcal{X}\cup\{v\}) - g(\mathcal{X}) \ge g(\mathcal{Y}\cup\{v\}) - g(\mathcal{Y}), \tag{14}$$

*which captures the diminishing marginal return characteristics such that the benefit of adding a new element into the set decreases as the set becomes larger.*

We now prove that the objective function of the problem **SUB** in Eq. (12) is monotone submodular.

**Proposition 1.** *The objective function in Eq. (12a) is a monotone submodular function over the ground set $\mathcal{V}$ as defined in Eq. (9).*

*Proof:* This proposition can be proved by using the definition of monotonicity and submodularity. ∎

We further observe the cache storage constraint of edge server $s \in \mathcal{S}$ in Eq. (12b), and note that each element $v^s_{f,m} \in \mathcal{A}$ (corresponding to the case $a^s_{f,m} = 1$ in $\boldsymbol{A}_{F\times M\times S}$) has a non-uniform

cost of $R_{f,m} \cdot T$ and $s$ has a storage budget of $B_s$. This constraint can be viewed as a knapsack constraint on the subset $\mathcal{V}_s \in \mathcal{V}$. Overall, the distributed caching placement problem in Eq. (12) is a submodular maximization problem subject to a set of knapsack constraints, which still is generally NP-hard and requires exponential computational complexity to reach the optimum by either ILP or other optimization methods. It is expected that by exploiting submodularity, the polynomial-time greedy algorithm is able to provide an effective approximation of the optimal solution of this NP-hard problem [41]. However, according to [41], [42], the greedy algorithm can only efficiently address the simplest case (i.e., a submodular maximization problem subject to one knapsack constraint) with theoretical approximation guarantee. When the number of knapsack constraints becomes greater than one, the greedy algorithm in general is no longer efficient, and in the worst case its approximation ratio will be arbitrarily bad. An exception exists if the set of multiple knapsack constraints form a matroid [38], such as the cache placement problem in [6] and [12] where the knapsack constraints are proved to be a partition matroid since all video files have the same size. In comparison, the proof of matroid for the multiple knapsack constraints in [6] and [12] no longer holds in our case because of the different video file sizes introduced by adaptive streaming. However, due to the special structure of the knapsack constraints in Eq. (12) (i.e., each knapsack constraint is imposed on the subset $\mathcal{V}_s \in \mathcal{V}$, and the set of all knapsack constraints is imposed on the finite ground set $\mathcal{V}$), we develop in the next subsection a polynomial-time greedy algorithm and provide a theoretical proof on the approximation ratio of the proposed greedy algorithm.

### C. Approximation Algorithm

To efficiently solve the submodular maximization problem in Eq. (12) with polynomial time complexity and theoretical approximation guarantees, we develop a $k$-cost benefit ($k$-CB) greedy algorithm. The system parameter, $k = 0, 1, 2, \ldots$ specifies the size of the initial set. Specifically, the proposed $k$-CB greedy algorithm considers all feasible initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$ of cardinality $k$. Starting from any initial set $\mathcal{A}^0$, at step $t$, the cost benefit greedy procedure iteratively searches over the remaining set $\mathcal{V}^{t-1} \setminus \mathcal{A}^{t-1}$ and inserts into the partial solution $\mathcal{A}^{t-1}$ an element according to Eqs. (16) and (17), until the remaining set reduces to an empty set. In other words, the cost benefit procedure adds at each iteration an element that maximizes the ratio between marginal benefit $Q(\mathcal{A}^{t-1} \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^{t-1})$ and cost $R_{f,m} \cdot T$ among all elements still affordable under the remaining storage budget until no more elements can be added. The proposed $k$-CB greedy algorithm then enumerates all initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$ of cardinality $k$, augments each of them following the cost benefit greedy procedure, and selects the initial set achieving the largest value of the objective function $Q(\mathcal{A}) = \sum_{u \in \mathcal{U}} Q_u(\mathcal{A})$ and finds its solution set as the final placement set $\mathcal{A}^*_k$. For the special case of $k = 0$, the algorithm reduces to a simple cost benefit greedy algorithm starting with $\mathcal{A}^0 = \emptyset$. On the other hand, if we remove the cost term $R_{f,m} \cdot T$ in Eqs. (15) and (16) and only add at each iteration an element maximizing the marginal benefit $Q(\mathcal{A}^{t-1} \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^{t-1})$, the algorithm reduces to a $k$-simple greedy algorithm. The complete $k$-cost benefit greedy algorithm is described in Algorithm 1. Since the $k$-simple greedy algorithm is only slightly different from Algorithm 1, it is thus omitted due to the space limit.

In terms of computational complexity, the running time of the proposed $k$-CB greedy algorithm is $O((SFM)^{k+1}U)$, indicating a polynomial time complexity and a very short additional implementation delay that is introduced by running the algorithm to find the final caching placement set. As the value of $k$ increases, the running time of the proposed algorithm becomes longer while the performance improves. In Theorem 1, we prove that when $k = 2$, the theoretical worst-case performance guarantee of the proposed algorithm is $\frac{1}{2}(1 - 1/e)$, i.e., its solution achieves at least the ratio $\frac{1}{2}(1 - 1/e) \approx 0.316$ of the optimal objective value. In practice, as it will be shown in the simulation results in Section VI, the algorithm performance approximation ratio is much higher than the theoretical lower bound, which is generally above 0.95.

**Algorithm 1** $k$-Cost benefit ($k$-CB) greedy algorithm

**Input:** system parameter $k$; finite ground set $\mathcal{V}$; video length $T$; encoding bitrate $R_{f,m}$ for any representation $z_{f,m} \in \mathcal{Z}$; and cache storage capacity $B_s$ for any edge server $s \in \mathcal{S}$.
**Output:** caching placement set $\mathcal{A}^*_k$

$id := 1$ // the index of the initial set
**for** any initial set $\mathcal{A}^0 \subseteq \mathcal{V}$ and $|\mathcal{A}^0| = k$ **do**
  $\mathcal{V}^0 := \mathcal{V}$ and $t := 1$ // initialization
  **for** $t = 1, 2, 3, \ldots$ **do**
    // greedy search iteration

$$\theta_t := \max_{v^s_{f,m} \in \mathcal{V}^{t-1} \setminus \mathcal{A}^{t-1}} \frac{Q(\mathcal{A}^{t-1} \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^{t-1})}{R_{f,m} \cdot T} \quad (15)$$

$$v^{s_t}_{f_t,m_t} := \arg \max_{v^s_{f,m} \in \mathcal{V}^{t-1} \setminus \mathcal{A}^{t-1}} \frac{Q(\mathcal{A}^{t-1} \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^{t-1})}{R_{f,m} \cdot T} \quad (16)$$

    **if**

$$\sum_{f \in \mathcal{F}} \sum_{m=1}^{M} \mathbf{1}|_{v^{s_t}_{f,m} \in (\mathcal{A}^{t-1} \cap \mathcal{V}_{s_t}) \cup \{v^{s_t}_{f_t,m_t}\}} \cdot R_{f,m} \cdot T \leq B_{s_t} \quad (17)$$

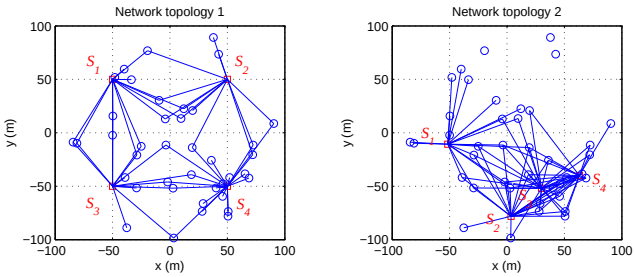
    **then**
      $\mathcal{A}^t := \mathcal{A}^{t-1} \cup \{v^{s_t}_{f_t,m_t}\}$ and $\mathcal{V}^t := \mathcal{V}^{t-1}$
    **else**
      $\mathcal{A}^t := \mathcal{A}^{t-1}$ and $\mathcal{V}^t := \mathcal{V}^{t-1} \setminus \{v^{s_t}_{f_t,m_t}\}$
    **end if**
    **if** $\mathcal{V}^t \setminus \mathcal{A}^t \neq \emptyset$ **then**
      $t := t + 1$
    **else**
      break
    **end if**
  **end for**
  $\mathcal{A}_{id} := \mathcal{A}^t$ and $id := id + 1$
**end for**
$\mathcal{A}^*_k := \arg\max_{i \in \{1,2,\ldots,id-1\}} \sum_{u \in \mathcal{U}} Q_u(\mathcal{A}_i)$

**Theorem 1.** *The better cache placement result achieved by running separately and comparing the 2-cost benefit greedy algorithm given in Algorithm 1 and the 2-simple greedy algorithm provides a $\frac{1}{2}(1 - 1/e)$ approximation. That is, in the worst case, it can achieve a performance guarantee of ratio $\frac{1}{2}(1 - 1/e)$ to the optimum.*

*Proof:* This theorem can be proved by using the diminishing return property of submodular functions. For the details, please refer to Appendix A. ■

## VI. Experimental Results

In this section, we evaluate the performance of our mobile edge caching placement optimization algorithms, and derive simple guidelines for effective cache allocation in wireless adaptive streaming systems under different simulation settings. We compare their performance with two schemes in the recent literature: 1) *Femto-Cache*, the femto-caching system and its associated greedy algorithm proposed in [12], which aims at minimizing the average downloading delay




(a) (b)

Fig. 2. Network connectivity graph with $S = 4$ edge servers and $U = 40$ independently and randomly distributed users, where (a) the edge servers are uniformly placed, and (b) the edge servers are placed according to the user distribution, i.e., more edge servers are placed in the area with higher user density.

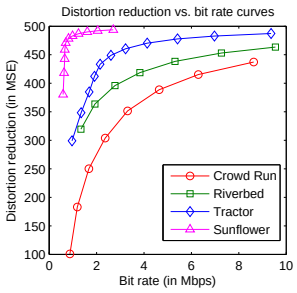



Fig. 3. Distortion reduction vs. encoding bitrate curves of the four videos.

of users for wireless video content delivery through allocating the cached content among the distributed edge servers; and 2) *Pop-Cache*, the popularity based adaptive streaming caching system proposed in [20], where each edge server caches all bitrate representations for a few top popular videos and only pre-fetches the highest bitrate representations for some other less popular videos, subject to the allocated cache storage capacity. As comparison algorithms, pop-cache is selected as a non-cooperative caching scheme that only considers the video content popularity, while femto-cache is chosen as a cooperative caching scheme that considers both the video content popularity and the cooperation among edge servers. The reason for this selection is to show the caching performance gain achieved by exploiting the cooperation through the comparison between pop-cache and femto-cache. In addition, as a comparison between the two cooperative caching schemes, the proposed algorithm achieves additional caching performance gain compared to femto-cache.

### A. Simulation Settings

We consider a wireless network where $U = 40$ users are randomly distributed in a cellular region formed by a disk of radius 100 m with the base station located at the center. Four edge servers ($S = 4$) are distributed in the cellular region in two different ways. Specifically, the edge servers are either uniformly placed as shown in Fig. 2(a), or placed according to the user distribution as shown in Fig. 2(b). The connectivity range (effective transmission range) of each edge server is set to 75 m, which results in the network connectivity graphs shown in Figs. 2(a) and 2(b). In accordance with the simulation settings in [6], we assume that the base station operates on a 20 MHz band with a spectral efficiency of 4 bits/s/Hz, while each edge server operates on a 20 MHz band with a spectral efficiency of 6 bits/s/Hz and the interference issue between the edge servers is neglected. Since current 802.11 WiFi standards allow operations on multiple 20 MHz bands, we further assume that the neighboring edge servers are operating on the orthogonal bands and each edge server allocates its transmission resource in a fair and uniform way between users.

Four test videos ($F = 4$, *Crowd Run*, *Riverbed*, *Tractor*, and *Sunflower*) with 1080p resolution ($1920 \times 1080$) [43] are selected as the video files needed for caching. These four test videos correspond to different content types, i.e., dense object motion for *Crowd Run*, rich details/fine textures and dense object motion for *Riverbed*, camera movement and medium object motion for *Tractor*, and small object motion for *Sunflower*, respectively. Suppose that the time duration of each video clip is $T = 10$ s, and $\Delta T = 1$ s is the time fraction within a video clip that is required to be buffered by the user before the actual playback starts on the user's screen, and the constant maximal distortion is set as $D_{\max} = 500$. At a frame rate of 30 fps, we further encode each video into $M = 3$ representations with encoding rate being $\{3R,\ 2R,\ R\}$ and $R = 2$ Mbps. The distortion reduction versus encoding bitrate curves of these four videos are illustrated in Fig. 3, where we see that the video content plays a key role in the rate-distortion characteristics. In particular, the distortion reduction increases faster with the rate when the video content has smaller motion. The storage capacity for each edge server is set to $B_s = 6RT = 120$ Mbits. We further assume that the popularity of the four videos follows a Zipf distribution with parameter 0.56 [30], i.e., the requesting probabilities of *Crowd Run*, *Riverbed*, *Tractor*, and *Sunflower* videos are 0.38, 0.25, 0.20, and 0.17, respectively [5]. We implement the proposed and comparison algorithms on a 48-processor server with 252 GB of RAM using Linux 3.1 kernel, where each processor is an Intel Xeon CPU E5-2680 at a clock frequency of 2.50 GHz.

### B. Performance Comparison

In Table II, we compare the performance of the different cache allocation algorithms in the two network topologies shown in Fig. (2), in terms of the theoretical computational complexity, average distortion reduction per user (achieved by the cached content in edge servers), approximation ratio with respect to the average distortion reduction per user, and base station transmission rate. Besides the proposed $k$-CB greedy algorithm, the simple greedy algorithm in Table II stands for the 0-simple greedy algorithm, where we remove the cost term $R_{f,m} \cdot T$ in Eqs. (15) and (16) and only add at each iteration an element maximizing the marginal benefit $Q(\mathcal{A}^{t-1} \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^{t-1})$ in Algorithm 1. In addition, the optimal solution of the **ILP** in Eq. (8) obtained by the IBM ILOG CPLEX solver [35] using a branch and bound method with a very high (i.e., exponential) time complexity $O(2^{2U \cdot 3F \cdot 3M \cdot 2S})$ is given as a performance upper bound. From the perspective of computational complexity, this optimal solution would become infeasible with the increase of either the number of representations or the network scale. In contrast, in different network topologies, the proposed $k$-CB greedy algorithm achieves a good approximation performance with the approximation ratio generally above 0.95 but with a much lower (i.e., polynomial) time complexity $O((SFM)^{k+1}U)$. The computational complexity of the proposed $k$-CB greedy algorithm could be further reduced as $k$ decreases, with the cost of only a slight reduction on the approximation ratio. Specifically, when $k = 0$, the proposed algorithm achieves a linear time complexity which is the same as the femto-cache and pop-cache algorithms.

As a performance comparison, the average distortion reduction per user and the approximation ratio achieved by the proposed $k$-CB greedy algorithm generally outperforms the other two comparison algorithms (femto-cache and pop-cache), while the base station

[5]Please note that this popularity distribution is chosen as an illustrative example. The proposed algorithm can be applied to any other popularity distribution, which is also experimentally justified in Table III in Section VI-D.

TABLE II
COMPARISON ON COMPUTATIONAL COMPLEXITY AND ALGORITHM PERFORMANCE

| Algorithm | Theoretical computation complexity | Network topology 1 | | | Network topology 2 | | |
|---|---|---|---|---|---|---|---|
| | | Ave. distortion reduction | Approx. ratio | BS rate (Mbps) | Ave. distortion reduction | Approx. ratio | BS rate (Mbps) |
| Optimum | $O(2^{2U \cdot 3F \cdot 3M \cdot 2S})$ | 423.0 | - | 0 | 399.7 | - | 6.00 |
| 3-CB Greedy | $O((SFM)^4U)$ | 418.6 | 0.990 | 0 | 397.5 | 0.995 | 6.00 |
| 2-CB Greedy | $O((SFM)^3U)$ | 416.6 | 0.985 | 0 | 397.0 | 0.993 | 6.34 |
| 1-CB Greedy | $O((SFM)^2U)$ | 409.9 | 0.969 | 0 | 393.9 | 0.986 | 6.69 |
| 0-CB Greedy | $O(SFMU)$ | 402.5 | 0.952 | 0 | 389.9 | 0.976 | 6.00 |
| Simple Greedy | $O(SFMU)$ | 344.4 | 0.814 | 17.98 | 356.9 | 0.893 | 14.72 |
| Femto-Cache | $O(SFMU)$ | 404.0 | 0.955 | 0 | 381.1 | 0.954 | 6.00 |
| Pop-Cache | $O(SFM)$ | 265.5 | 0.628 | 29.88 | 308.4 | 0.772 | 22.75 |

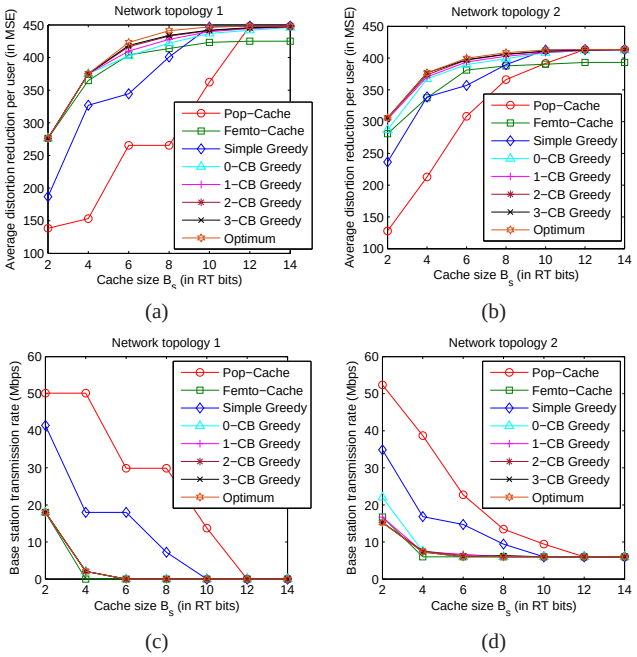



Fig. 4. Average distortion reduction per user vs. cache size in (a) network topology 1, and (b) network topology 2; and base station transmission rate vs. cache size in (c) network topology 1, and (d) network topology 2; where the simulation setting is $F = 4$ video files, $M = 3$ representations, $S = 4$ edge servers, and $U = 40$ users.

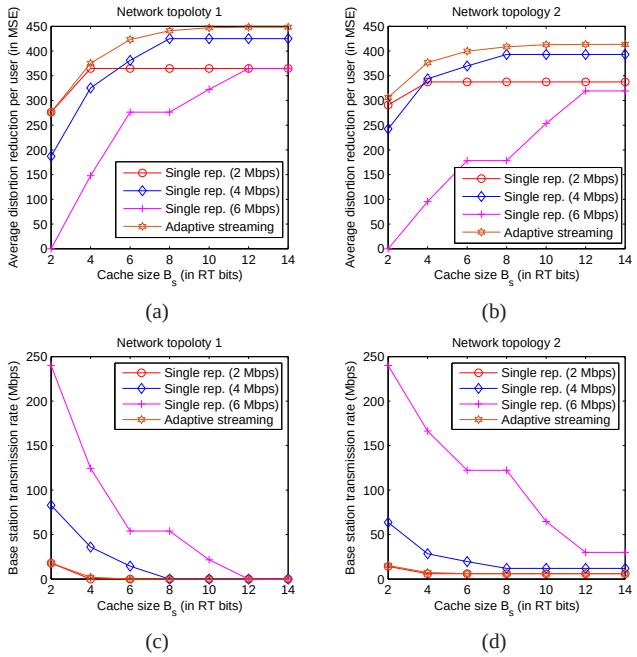



Fig. 5. Comparison between multiple representation caching and single representation caching: average distortion reduction per user vs. cache size in (a) network topology 1, and (b) network topology 2; and base station transmission rate vs. cache size in (c) network topology 1, and (d) network topology 2; where the simulation setting is $F = 4$ video files, $M = 3$ representations, $S = 4$ edge servers, and $U = 40$ users.

transmission rate incurred by the proposed algorithm is usually kept at a very low level. For example, pop-cache and femto-cache algorithms result in 0.772- and 0.954-approximation ratio of the optimal solution in network topology 2, with the base station transmission rate of 22.75 Mbps and 6.00 Mbps, respectively. For the case of $k = 0$ and $k = 1$, the proposed $k$-CB greedy algorithm advances the approximation ratio to 0.976 and 0.986, respectively. When $k$ becomes large, e.g. $k = 3$, the proposed algorithm can even achieve 0.995-approximation ratio, while the base station transmission rate is 6 Mbps, which is the same as the optimal solution. The fundamental reason why the proposed algorithm outperforms the others is the following. In addition to the consideration of video file popularity and the cooperation among different edge servers, the caching decision for the representations of different videos can be further adapted to the video content characteristics in our algorithm. For videos with small motion (e.g., *Tractor* and *Sunflower*), the proposed algorithm only allocates the basic representation with the smallest bitrate $R$ at each edge server, while for videos with larger motion (e.g., *Crowd Run* and *Riverbed*), representations with larger bitrate $2R$ or $3R$ are allocated at some edge servers to gain larger distortion reduction. As a result, a better overall cache allocation performance can be achieved by the proposed algorithm.

### C. Impact of System Parameters

In this subsection, we evaluate and compare the algorithm performance of different schemes under various simulation settings, in order to gain a further insight into the impact of different system parameters. In this set of simulations, we still adopt the same settings as in Section VI-A, unless stated otherwise.

*1) Cache Size $B_s$:* Fig. 4 illustrates the results measuring the average distortion reduction per user and the additional base station transmission rate under two different network topologies, by varying the cache size of each edge server $B_s$. In this simulation, all edge servers have the same cache size, which is varied from $2RT = 40$ Mbits to $14RT = 280$ Mbits. The general observation for all algorithms under different network topologies is that the average distortion reduction per user increases and the base station transmission rate decreases, as the cache size gradually increases. The reason is that, the edge server can pre-fetch more video representations in its local cache with the increment of the cache size, which in turn can create more opportunities for the different edge servers to serve more user requests without the need to communicate with the base station. For the comparison algorithms, when the cache size is small (e.g., $B_s = 2RT$) such that a very limited number of representations could be stored in the edge servers, the femto-cache algorithm outperforms

the pop-cache algorithm with the achieved average distortion reduction per user very close to the optimal solution. On the contrary, when the cache size is large (e.g., $B_s = 12RT$), the pop-cache algorithm outperforms the femto-cache algorithm in terms of the achieved average distortion reduction per user. Compared with these two algorithms, the proposed $k$-CB greedy algorithm can achieve a better approximation performance in terms of the largest average distortion reduction per user (i.e., largest approximation ratio), while the additional base station transmission rate is very close to the optimal solution. The simple greedy algorithm stands for the 0-simple greedy algorithm that is obtained by removing the caching storage cost consideration (i.e., the term $R_{f,m} \cdot T$ in Eqs. (15) and (16)) in Algorithm 1 and setting $k = 0$. Therefore, the average distortion reduction per user achieved by the simple greedy algorithm is comparable to the proposed $k$-CB greedy algorithm when the caching storage resource is not limited (i.e., when the cache size $B_s$ is large). In contrast, when $B_s$ is small, the overall caching performance of the simple greedy algorithm is even poorer than femto-cache algorithm. In addition, for all different cache sizes $B_s$ and different network topologies, the average distortion reduction per user will be improved with the increment of the initial set size $k$. It can be seen in Fig. 4 that when $k$ increases to 3, all the performance curves almost overlap with those of the optimal solution.

For the same setting as in Fig. 4, we further show in Fig. 5 the additional caching performance gain introduced by adaptive streaming with multiple representations for each video, in terms of both the average distortion reduction per user and the base station transmission rate. Here, adaptive streaming method denotes the optimal solution to the **ILP** in Eq. (8), where each video is encoded to three representations with encoding rate being 6 Mbps, 4 Mbps and 2 Mbps, respectively. In comparison, single rep. (e.g., 2 Mbps) method represents the optimal solution to the **ILP** in Eq. (8), where only single representation is encoded for each video with a specific encoding rate (e.g., 2 Mbps). It can be demonstrated that for both network topologies and different caching sizes, a higher average distortion reduction per user and a lower base station transmission rate can be achieved through introducing adaptive streaming into the caching system, compared to any of the three single representation caching cases.

*2) Number of Users $U$:* From the result shown in Table II and Figs. 4 and 5, it is justified that the performance comparison among different algorithms is similar for both network topology 1 and network topology 2, i.e., independent of the specific network topology. Therefore, we select network topology 1 shown in Fig. 2(a) as the representative network in the following subsections, and studied the impact of other parameters. We vary the number of users, and accordingly shown in Fig. 6 the average distortion reduction per user, base station transmission rate and algorithm running time achieved by different algorithms under two cache size settings, namely $B_s = 6RT = 120$ Mbits and $B_s = 10RT = 200$ Mbits, respectively. Figs. 6(a) and 6(b) show that the average distortion reduction per user generally decreases as the number of users increases. The reason is that the base station and all the edge servers allocate their transmission resources fairly to all the connected users. When more users join the network and connect to the base station and edge servers, they will compete for the shared transmission resources, which indicates a higher probability of communication link interference and lowers the average user throughput. The major exception occurs when the number of users $U$ is small for the pop-cache algorithm. For example, instead of the expected decreasing behavior, the average distortion reduction per user achieved by pop-cache algorithm would increase when $U$ increases from 40 to 60 in Fig. 6(a). This can be explained as follows. When the local cache

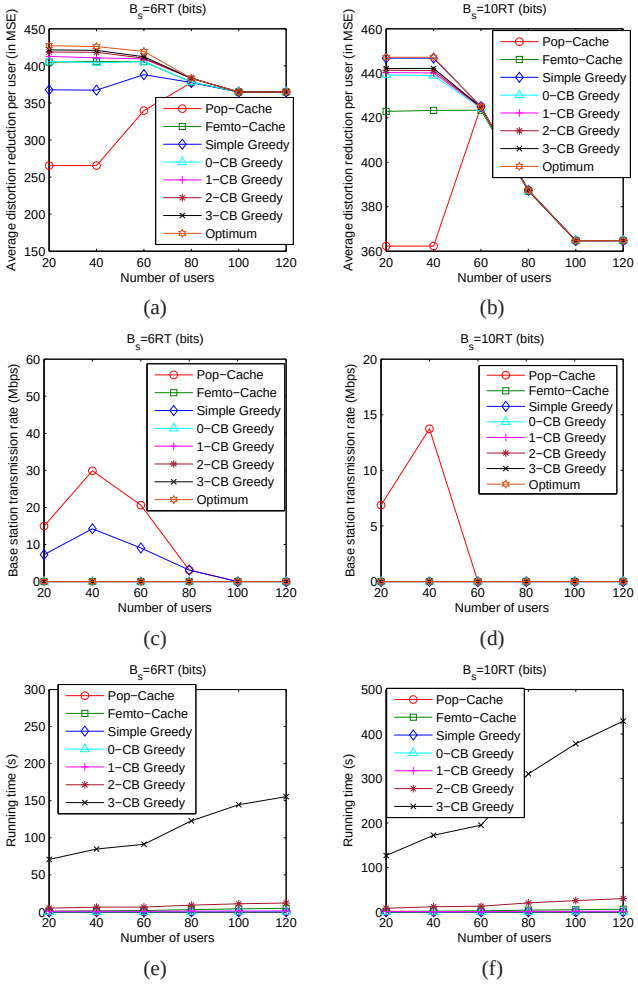



Fig. 6. (a) Average distortion reduction per user, (c) base station transmission rate and (e) algorithm running time vs. number of users when cache size $B_s = 6RT = 120$ Mbits; and(b) average distortion reduction per user, (d) base station transmission rate and (f) algorithm running time vs. number of users when cache size $B_s = 10RT = 200$ Mbits; where the simulation is conducted in network topology 1 with $F = 4$ video files, $M = 3$ representations, and $S = 4$ edge servers.

size is limited, pop-cache algorithm allocates as many representations with the highest bitrate as possible to each edge server while the caching priority of each representation is in a decreasing order of the popularity. For $U$=40, the average user throughput in the network allows the edge server to pre-fetch the highest bitrate representation of 6 Mbps. When $U$ increases to 60, the highest bitrate representation allowed to be cached in the edge server reduces to 4 Mbps due to the reduction of the average user throughput. Therefore, more representations of different videos with lower bitrates can be cached in each edge server, which in turn results in a higher distortion reduction for each user. It can also be noted that when the cache size is large enough (the $B_s = 10RT$ case) to pre-fetch a large number of representations, the simple greedy algorithm could achieve the same average distortion reduction per user as the optimal solution. Figs. 6(c) and 6(d) show that the base station rate achieved by the proposed $k$-CB greedy algorithm is the same as the optimal solution, which is 0 for different number of users.

The algorithm running time is another performance metric which has the same as the average distortion reduction per user. In Figs. 6(e) and 6(f), we compare the actual running time of different algorithms, and show the impact of the number of users $U$ on the running time. Through the curves in Figs. 6(e) and 6(f), the previous theoretical analysis of the computational complexity is well justified. That is,

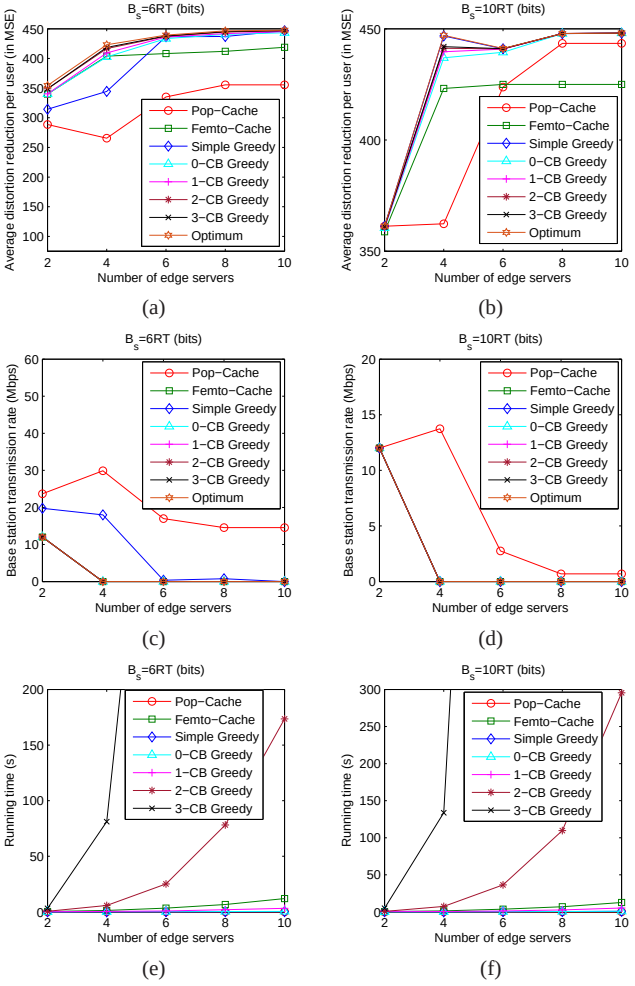



Fig. 7. (a) Average distortion reduction per user, (c) base station transmission rate and (e) algorithm running time vs. number of edge servers when cache size $B_s = 6RT = 120$ Mbits; and (b) average distortion reduction per user, (d) base station transmission rate and (f) algorithm running time vs. number of edge servers when cache size $B_s = 10RT = 200$ Mbits; where the simulation is conducted in network topology 1 with $F = 4$ video files, $M = 3$ representations, and $U = 40$ users.

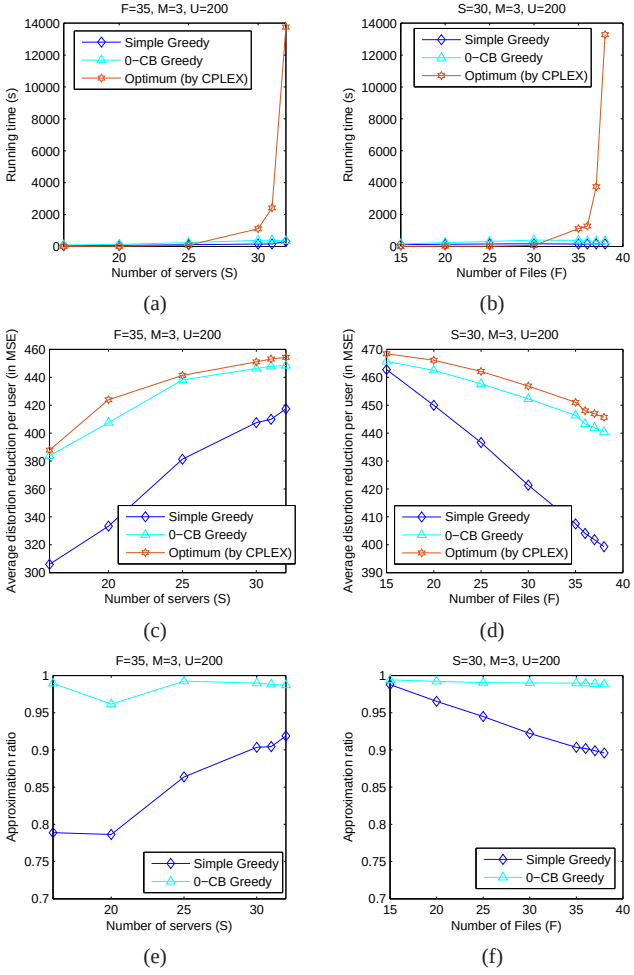



Fig. 8. (a) Algorithm running time, (c) average distortion reduction per user, and (e) approximation ratio vs. number of edge servers, and (b) algorithm running time, (d) average distortion reduction per user, and (f) approximation ratio vs. number of video files, where the video popularity distributions follow a Zipf distribution with parameter 0.8.

the proposed $k$-CB greedy algorithm and simple greedy algorithm, as well as the other two comparison algorithm, have the polynomial computational complexity as shown in Table II. Specifically, the computational complexity of all the algorithms (excluding the pop-cache algorithm) is linear with respect to $U$, while the computational complexity of the pop-cache algorithm is not affected by $U$.

*3) Number of Edge Servers $S$:* We still consider a network organization as in the network topology 1 shown in Fig. 2(a), but vary the number of edge servers that are uniformly placed in the cellular region. We then show in Fig. 7 the average distortion reduction per user, base station transmission rate and algorithm running time achieved by different algorithms under two cache size settings $B_s = 6RT = 120$ Mbits and $B_s = 10RT = 200$ Mbits, respectively. Figs. 7(a) and 7(b) show that the average distortion reduction per user generally increases as we place more edge servers in the cellular region. The reason is that when the number of edge servers increases, each edge server serves a smaller number of users, which decreases the probability of communication link interference among users and thus increases the average user throughput. In addition, a denser deployment of the edge servers within the same cellular region creates more opportunities for the coordination between edge servers to cache different representations and better support the users' requests by the cached content. Figs. 7(c) and 7(d) show that the base station rate achieved by the proposed $k$-CB greedy algorithm is the same as the optimal solution.

Figs. 7(e) and 7(f) illustrate the comparison of the algorithm running time versus the number of edge servers $S$ achieved by the different algorithms. These actual running time curves also justify that the computational complexity of all the algorithms have the polynomial computational complexity as shown in Table II.

### *D. Performance Evaluation for Larger System Settings*

Finally, we conduct simulations for larger scale settings. In total, $F = 15$ test videos with 1080p resolution ($1920 \times 1080$) [6], available at [43], are selected as the video files to be cached in the edge servers and requested by the users. They correspond to different motion and video types (such as, sports, documentary, cartoon and movie). For the video popularity, we investigate three different popularity distributions, i.e., the Zipf distribution with parameter 0.8 and 0.56, and the uniform distribution. We also consider a larger wireless network with $U = 200$ users randomly distributed in a cellular region formed by a disk of radius 200 m, and $S = 16$ edge servers uniformly placed in this cellular region. The storage capacity of each edge server is set to $B_s = 18RT = 360$ Mbits, and all the other parameters are

[6]These videos are: *Aspen*, *Blue Sky*, *Controlled Burn*, *Crowd Run*, *Dinner*, *Ducks Take Off*, *Riverbed*, *In To Tree*, *Life*, *Old Town Cross*, *Station2*, *Sunflower*, *Touchdown Pass*, *Tractor*, and *Park Joy*.

TABLE III
COMPARISON OF AVERAGE DISTORTION REDUCTION PER USER AND BASE STATION TRANSMISSION RATE UNDER DIFFERENT POPULARITY DISTRIBUTIONS.

| Algorithm | Zipf distribution, parameter 0.8 | | Zipf distribution, parameter 0.56 | | Uniform distribution | |
|---|---|---|---|---|---|---|
| | Ave. distortion reduction | BS rate (Mbps) | Ave. distortion reduction | BS rate (Mbps) | Ave. distortion reduction | BS rate (Mbps) |
| Optimum | 459.2 | 1.63 | 455.2 | 0.45 | 449.4 | 0 |
| 2-CB Greedy | 454.4 | 1.34 | 450.1 | 1.77 | 443.0 | 2.4 |
| 1-CB Greedy | 454.2 | 1.34 | 449.9 | 1.77 | 442.6 | 2.4 |
| 0-CB Greedy | 453.6 | 1.34 | 449.4 | 1.77 | 442.0 | 2.4 |
| Simple Greedy | 408.4 | 52.69 | 394.1 | 62.35 | 371.7 | 76.67 |
| Femto-Cache | 442.2 | 1.31 | 436.3 | 1.65 | 426.9 | 2.4 |
| Pop-Cache | 379.2 | 75.41 | 350.7 | 97.22 | 284.2 | 150.27 |

the same as previously. The other simulation settings are the same as in Section VI-A.

In Table III, we compare the average distortion reduction per user and the base station transmission rate obtained by different caching placement algorithms under the three different popularity distributions. Although the system settings scale with a larger number of videos, edge servers and users, it is again verified that, for all popularity distributions the proposed $k$-CB greedy algorithm outperforms the femto-cache and pop-cache algorithms. It achieves a higher average distortion reduction per user, and comparable or lower base station transmission rate. Specifically, for all popularity distributions, the proposed $k$-CB greedy algorithm ($k = 0, 1, 2$) improves the average distortion reduction per user by at least 11.4 (in MSE) compared to the femto-cache algorithm, and improves by at least 74.4 (in MSE) compared to the pop-cache algorithm. This average video distortion reduction per user performance is only about 5 (in MSE) lower than the optimal solution, while the additional base station transmission rate is comparable with the optimal solution. In terms of the average additional base station transmission rate over all the three popularity distributions, the difference between the proposed $k$-CB greedy algorithm and the optimal solution is only 1.1 Mbps.

In order to gain a further insight into the superiority of the proposed algorithm over the optimal ILP solution provided by the generic solver IBM ILOG CPLEX [35], we compare the performance of the proposed 0-CB greedy algorithm, the simple greedy algorithm, and the optimal solution in Fig. 8. Specifically, we show the performance comparison of the algorithm running time, the average distortion reduction per user, and the approximation ratio versus the number of edge servers $S$ in Figs. 8(a)-8(c), respectively. The same set of performance comparison versus the number of video files $F$ is shown in Figs. 8(d)-8(f), respectively. We see that the previous theoretical analysis of the computational complexity is well justified. That is, the optimal solution needs a very high computational complexity which is exponential to $S$ and $F$, while both the proposed 0-CB greedy algorithm and simple greedy algorithm achieve a linear computational complexity. In addition, the overall approximation ratio of the proposed 0-CB greedy algorithm is greater than 0.95 in Fig. 8(c) and greater than 0.99 in Fig. 8(f), respectively. Therefore, the performance of the proposed 0-CB greedy algorithm is very close to the performance upper bound guided by the optimal solution, but the actual running time is much shorter. In other words, the proposed algorithm has a much lower increasing rate of the running time and scales better than the optimal solution solved by the generic solver IBM ILOG CPLEX [35]. Considering a practical wireless video caching system with a large number of videos, representations, edge servers and users, the long waiting time for the IBM ILOG CPLEX solver to obtain the optimal solution makes it infeasible in practice. In contrast, the proposed algorithm is suitable for the delay sensitive video applications since it is capable of achieving a near-optimal solution within a short period of time.

Next, in Fig. 9, we proportionally scale up the system according to the settings in Table IV and show its impact on different algorithms.

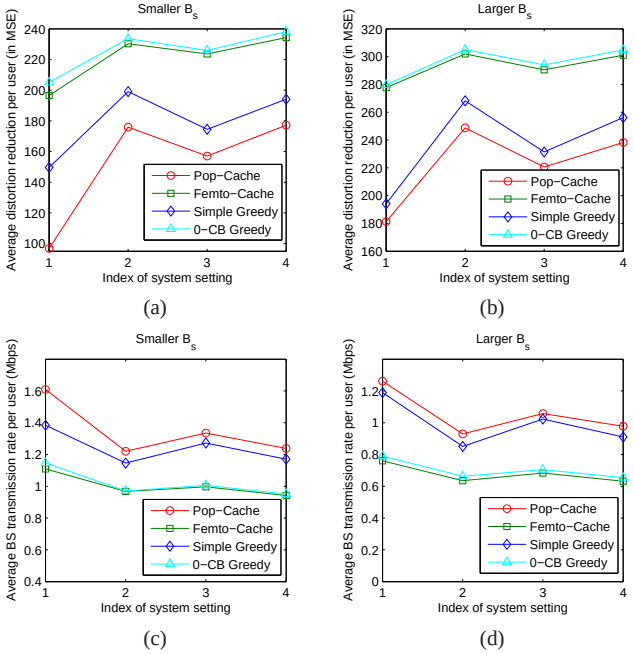



Fig. 9. (a) Average distortion per user and (c) average base station transmission rate per user vs. system scale when cache size $B_s$ is set to $30RT = 12G$ bits, $60RT = 24G$ bits, $90RT = 36G$ bits, and $120RT = 48G$ bits, respectively, for the four system settings; and (b) Average distortion per user and (d) average base station transmission rate per user vs. system scale when cache size $B_s$ is set to $60RT = 24G$ bits, $120RT = 48G$ bits, $180RT = 72G$ bits, and $240RT = 96G$ bits, respectively, for the four system settings. The video popularity distributions follow a Zipf distribution with parameter 0.8.

Here, we assume that each edge server operates on a 20 MHz band with a spectral efficiency of 60 bits/s/Hz, and the length of each video clip is $T = 200$ s. The other simulation settings are the same as in Section VI-A. For these large system settings, it becomes infeasible for IBM ILOG CPLEX solver to get the optimal solution due to the exponential computation complexity issue. Therefore, we compare the caching performance (in terms of the average distortion reduction per user and the average base station transmission rate per user) of the proposed 0-CB greedy algorithm with the simple greedy, femto-cache and pop-cache algorithms in Fig. 9. It can be seen that scaling up the system will not degrade the caching performance of all the different algorithms. In particular, the proposed 0-CB greedy algorithm keeps a relatively stable caching performance and a relatively stable performance gain compared to other algorithms under all the different system settings, which indicates that it can also be applied to VoD systems with larger settings than any of the system settings studied in this paper.

### E. Discussion

In terms of the system design, the above observations show that, when the cache size of each edge server is large enough to pre-

TABLE IV
SYSTEM SETTINGS IN FIG. 9.

| Index | No. of videos $F$ | No. of users $U$ | No. of edge servers $S$ | Radius of cellular region | Smaller cache size $B_s$ | Larger cache size $B_s$ |
|---|---|---|---|---|---|---|
| 1 | 250 | 500 | 4 | 200 m | $30RT = 12G$ bits | $60RT = 24G$ bits |
| 2 | 500 | 1000 | 8 | 280 m | $60RT = 24G$ bits | $120RT = 48G$ bits |
| 3 | 750 | 1500 | 12 | 350 m | $90RT = 36G$ bits | $180RT = 72G$ bits |
| 4 | 1000 | 2000 | 16 | 400 m | $120RT = 48G$ bits | $240RT = 96G$ bits |

fetch a large number of representations, the proposed simple greedy algorithm could almost achieve the same average distortion reduction per user as the optimal solution, with only a linear computational complexity. For the proposed $k$-CB greedy algorithm, it generally outperforms the other comparison algorithms for different simulation settings. In addition, we can seek the tradeoff between the algorithm performance and the computational complexity (algorithm running time) by adapting the value of the initial set size $k$. A larger $k$ improves the algorithm's performance, but at the cost of a longer execution time. In practice, to have a near optimal approximation solution with affordable algorithm running time, we could set $k$ to 0 or 1 for large scale networks.

In addition, these observations could further provide some design guidelines for the edge servers to select the cached representations with corresponding bitrates. That is, the caching placement strategy for all the representations of all the videos is not only dependent on the video popularity distribution, but also affected by the video content characteristics (i.e., the R-D behavior). For the same video type, straightforwardly, a larger amount of representations with higher bitrates needs to be cached by the edge servers for more popular videos. While for different video types, a larger amount of representations with higher bitrates need to be cached by the edge servers for videos with larger motion of the objects, or videos with more complex content (e.g., dense objects, camera movement, and zoom effect). Overall, the proposed algorithm complies well with these design guidelines and scales well with the size of the system. Since it could further strike the tradeoff between the algorithm performance and the computational complexity (algorithm running time), it is therefore useful for the practical system design.

## VII. CONCLUSION

This paper studied a QoE-driven mobile edge caching placement optimization problem for adaptive streaming systems. We have provided an ILP formulation to achieve the performance upper bound, and an equivalent constrained submodular maximization that is used to develop an approximate algorithm with polynomial time complexity. Simulation results have justified that the proposed cost benefit greedy algorithm could achieve a near-optimal performance without introducing a long additional computation delay, which is therefore suitable for delay sensitive applications such as adaptive streaming. These results also demonstrated that by introducing adaptive streaming to allow caching multiple representations for the same video, the proposed caching placement optimization framework could achieve an additional caching performance gain (in terms of higher average distortion reduction per user and lower base station transmission rate) over the single-representation caching schemes. We also found that the performance of the caching placement is greatly affected by the R-D properties of different video contents, in addition to the common considerations (such as video content popularity distribution and network conditions) of existing caching schemes on adaptive streaming. Based on the analysis and simulation results, we further provided some design guidelines for the caching resource allocation of the edge servers among multiple bitrate representations. For the same video type, a larger amount of representations with higher bitrates needed to be cached by the edge servers for more popular videos. While for different video types, a larger amount of representations with higher bitrates needed to be cached by the edge servers for videos with larger motion of the objects, or videos with more complex content. For future work, we plan to formally extend the proposed mobile edge caching placement policy to future network architectures, such as information-centric networks (ICNs) and software-defined networks (SDNs).

## APPENDIX A
## PROOF OF THEOREM 1

If the cardinality of the optimal solution to the problem **SUB** in Eq. (12) is not greater than two, then such a solution can be found by Algorithm 1 through enumerating all possible sets with cardinality of two or less. In the following, we only consider the case that the optimal solution to problem **SUB** has a cardinality greater than two. Specifically, denote $\mathcal{A}^*$ as the optimal solution, which is further ordered such that:

$$Q(\{v_1, v_2, \ldots, v_t\}) = \max_{v \in \mathcal{A}^* \setminus \{v_1, v_2, \ldots, v_{t-1}\}} Q(\{v_1, v_2, \ldots, v_{t-1}\} \cup \{v\}). \tag{18}$$

In other words, $v_1$ is an element of the optimal solution set $\mathcal{A}^*$ that has the largest value of the objective function, and $v_2$ is an element that achieves the largest marginal increase in the value of the objective function if it is added to the set $\{v_1\}$, and so on. Denote $\mathcal{A}^{0*} = \{v_1, v_2\}$ as the set comprising the first two elements of the optimal solution set $\mathcal{A}^*$. For any element $v_k \in \mathcal{A}^* \setminus \mathcal{A}^{0*}$ and any set $\mathcal{Y} \subseteq \mathcal{V}$, following from the submodularity and the ordering property of the optimal solution set $\mathcal{A}^*$, we have the following inequalities:

$$\begin{aligned} Q(\mathcal{A}^{0*} \cup \mathcal{Y} \cup \{v_k\}) - Q(\mathcal{A}^{0*} \cup \mathcal{Y}) &\leq Q(\{v_k\}) - Q(\emptyset) \\ &\leq Q(\{v_k\}) \leq Q(\{v_1\}), \end{aligned} \tag{19}$$

$$\begin{aligned} Q(\mathcal{A}^{0*} \cup \mathcal{Y} \cup \{v_k\}) - Q(\mathcal{A}^{0*} \cup \mathcal{Y}) &\leq Q(\{v_1\} \cup \{v_k\}) - Q(\{v_1\}) \\ &\leq Q(\{v_1, v_2\}) - Q(\{v_1\}). \end{aligned} \tag{20}$$

By summing up Eqs. (19) and (20), we have:

$$2[Q(\mathcal{A}^{0*} \cup \mathcal{Y} \cup \{v_k\}) - Q(\mathcal{A}^{0*} \cup \mathcal{Y})] \leq Q(\mathcal{A}^{0*}). \tag{21}$$

Since the proposed cost-benefit greedy algorithm enumerates all possible choices of the starting set with cardinality of two, we consider a specific greedy procedure within Algorithm 1 where the set $\mathcal{A}^{0*}$ is selected as the starting set, i.e., $\mathcal{A}^0 = \mathcal{A}^{0*}$. Next, we will prove that the objective function value of the solution set obtained by this greedy procedure guarantees at least a ratio $\frac{1}{2}(1 - 1/e)$ to the value achieved by the optimal solution set $\mathcal{A}^*$.

Define a new set function $g(\mathcal{A}) = Q(\mathcal{A}) - Q(\mathcal{A}^{0*})$, and its monotone submodularity can be directly obtained since $Q(\mathcal{A})$ is a monotone submodular function as shown in Proposition 1 and $Q(\mathcal{A}^{0*})$ has a constant value. For any step $t$, we have:

$$g(\mathcal{A}^*) \leq g(\mathcal{A}^t \cup \mathcal{A}^*) = g(\mathcal{A}^t \cup (\mathcal{A}^* \setminus \mathcal{A}^t)) \tag{22a}$$

$$\leq g(\mathcal{A}^t) + \sum_{v^s_{f,m} \in \mathcal{A}^* \setminus \mathcal{A}^t} [g(\mathcal{A}^t \cup \{v^s_{f,m}\}) - g(\mathcal{A}^t)] \tag{22b}$$

$$= g(\mathcal{A}^t) + \sum_{v^s_{f,m} \in \mathcal{A}^* \setminus \mathcal{A}^t} [Q(\mathcal{A}^t \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^t)], \tag{22c}$$

where, Eqs. (22a) and (22b) follow from the monotonicity and submodularity of the set function $g(\mathcal{A})$, respectively; and Eq. (22c) is obtained by applying the definition of $g(\mathcal{A})$.

Let $t^*$ be the last step at which the greedy procedure can add a new element to $\mathcal{A}^{t^*-1}$, i.e., for $t > t^*$, the greedy procedure cannot add any new element to $\mathcal{A}^t$ due to the capacity constraint in Eq. (17). In this case, the approximate solution is obtained as $\mathcal{A}^{t^*}$. If $\mathcal{A}^{t^*} = \mathcal{A}^*$, then such an approximate solution achieves the optimal performance, otherwise, there must exist an element $v_{t'+1} \in \mathcal{A}^*$ which is obtained from Eq. (16) at step $t'$ but cannot be added to the set $\mathcal{A}^{t'}$ since $R_{f_{t'+1}m_{t'+1}} \cdot T + \sum_{f\in\mathcal{F}} \sum_{m=1}^{M} \mathbf{1}|_{v^{s_{t'}}_{f,m} \in (\mathcal{A}^{t'} \cap \mathcal{V}_{s_{t'+1}})} \cdot R_{f,m} \cdot T > B_{s_{t'+1}}$. Without loss of generality, we further assume that $t'_i+1,\ i = 1, 2, 3, \ldots, N_t$ denotes the ordered steps for all $v_{t'_i+1} \in \mathcal{A}^*$ but not added to $\mathcal{A}^{t'_i}$ until time $t$. Then, for all $t = 0, 1, 2, \ldots, t^*$, according to Eq. (22), we have the following inequality:

$$
\begin{aligned}
g(\mathcal{A}^*) \;\le\; & g(\mathcal{A}^t) + \sum_{v^s_{f,m}\in\mathcal{A}^*\setminus(\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}\cup\mathcal{A}^t)} R_{f,m}\cdot T\cdot\theta_{t+1} \\
& + \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1} && \text{(23a)} \\
\le\; & g(\mathcal{A}^t) + \Big(\sum_{s\in\mathcal{S}} B_s - \sum_{v^s_{f,m}\in\mathcal{A}^*\cap\mathcal{A}^t} R_{f,m}\cdot T\Big)\theta_{t+1} \\
& + \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1} && \text{(23b)} \\
\le\; & g(\mathcal{A}^t) + \Big(\sum_{s\in\mathcal{S}} B_s - \sum_{v^s_{f,m}\in\mathcal{A}^{0*}} R_{f,m}\cdot T\Big)\theta_{t+1} \\
& + \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1}. && \text{(23c)}
\end{aligned}
$$

In Eq. (23a), we divide the set $\mathcal{A}^* \setminus \mathcal{A}^t$ into $\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}$ (the subset of nodes that are in $\mathcal{A}^*$ but not added into $\mathcal{A}^t$) and $(\mathcal{A}^* \setminus \mathcal{A}^t) \setminus \bigcup_{i=1}^{N_t}\{v_{t'_i+1}\} = \mathcal{A}^* \setminus (\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\} \cup \mathcal{A}^t)$. Based on the update and determination procedure in Lines 8-12 in Algorithm 1, we have $(\mathcal{A}^* \setminus \bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}) \subseteq \mathcal{V}^t$ and thus $\mathcal{A}^* \setminus (\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\} \cup \mathcal{A}^t) \subseteq \mathcal{V}^t \setminus \mathcal{A}^t$. From Eq. (15), we then have $Q(\mathcal{A}^t \cup \{v^s_{f,m}\}) - Q(\mathcal{A}^t) \le R_{f,m}\cdot T\cdot\theta_{t+1}, \forall v^s_{f,m} \in \mathcal{A}^* \setminus (\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\} \cup \mathcal{A}^t)$. Next, we consider the nodes in $\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}$. Since each node $v_{t'_i+1}$ is obtained based on Eqs. (15) and (16) at step $t'_i$, we have $Q(\mathcal{A}^{t'_i} \cup \{v^{s_{t'_i}}_{f_{t'_i},m_{t'_i}}\}) - Q(\mathcal{A}^{t'_i}) = R_{f_{t'_i},m_{t'_i}}\cdot T\cdot\theta_{t'_i+1}$. Since $t'_i+1 \le t$ and $\mathcal{A}^{t'_i} \subseteq \mathcal{A}^t$, we have $Q(\mathcal{A}^t \cup \{v^{s_{t'_i}}_{f_{t'_i},m_{t'_i}}\}) - Q(\mathcal{A}^t) \le R_{f_{t'_i},m_{t'_i}}\cdot T\cdot\theta_{t'_i+1}$ from the submodularity. Therefore, the inequality in Eq. (23a) holds; Eq. (23b) follows from the fact that $\mathcal{A}^*$ is a feasible set and thus $\sum_{v^s_{f,m}\in\mathcal{A}^*} R_{f,m}\cdot T \le \sum_{s\in\mathcal{S}} B_s$, Eq. (23c) is obtained since $\mathcal{A}^{0*} \subseteq \mathcal{A}^t$.

Denote $\mathcal{T}_t$ as the set of time steps at which the element $v^{s_t}_{f_t,m_t}$ obtained by Eq. (16) can be added into $\mathcal{A}^{t-1}$. Let $W_t = \sum_{\tau\in\mathcal{T}^t} R_{f_\tau,m_\tau}\cdot T$ and $W_0 = 0$. By the definition of the element $v_{t^*}$, we denote $W' = W_{t^*+1} = W_{t^*} + R_{f_{t^*+1},m_{t^*+1}}\cdot T$ and have $W' \ge \sum_{s\in\mathcal{S}} B_s - \sum_{v^s_{f,m}\in\mathcal{A}^{0*}} R_{f,m}\cdot T = W''$. For $j = 1, 2, \ldots, W'$, we define an auxiliary variable $\rho_j = \theta_t$ if $j = W_{t-1}+1, \ldots, W_t$. Therefore, we have the following relationship between $\rho_j$ and $W_t$. For any $t \in \mathcal{T}_t$, $\sum_{j=W_{t-1}+1}^{W_t}\rho_j = (W_t - W_{t-1})\cdot\theta_t = R_{f_t,m_t}\cdot T\cdot\theta_t$; for any $t \notin \mathcal{T}_t$, we have $\sum_{j=W_{t-1}+1}^{W_t}\rho_j = 0$ since $W_t = W_{t-1}$. The definition of $\mathcal{T}_t$ is the set of time steps at which the element $v^{s_t}_{f_t,m_t}$ obtained by Eq. (16) can be added into $\mathcal{A}^{t-1}$. Since $t^*$ is the last step at which the greedy procedure can add a new element to $\mathcal{A}^{t^*-1}$, we have $t^* \in \mathcal{T}^{t^*}$, and

$$
\begin{aligned}
& g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) && \text{(24a)} \\
&= Q(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - Q(\mathcal{A}^{0*}) && \text{(24b)} \\
&= Q(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - Q(\mathcal{A}^{t^*}) + Q(\mathcal{A}^{t^*-1} \cup \{v_{t^*}\}) - Q(\mathcal{A}^{0*}) && \text{(24c)} \\
&= \cdots\cdots \\
&= Q(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - Q(\mathcal{A}^{t^*}) + \sum_{\tau\in\mathcal{T}^{t^*}} [Q(\mathcal{A}^{\tau-1} \cup \{v_\tau\}) - Q(\mathcal{A}^{\tau-1})] && \text{(24d)} \\
&= R_{f_{t^*+1},m_{t^*+1}}\cdot T\cdot\theta_{t^*+1} + \sum_{\tau\in\mathcal{T}^{t^*}} R_{f_\tau,m_\tau}\cdot T\cdot\theta_\tau && \text{(24e)} \\
&= (W_{t^*+1} - W_{t^*})\cdot\theta_{t^*+1} + \sum_{\tau\in\mathcal{T}^{t^*}} (W_\tau - W_{\tau-1})\cdot\theta_\tau && \text{(24f)} \\
&= \sum_{j=1+W_{t^*}}^{W_{t^*+1}} \rho_j + \sum_{j=1}^{W_{t^*}} \rho_j = \sum_{j=1}^{W'} \rho_j. && \text{(24g)}
\end{aligned}
$$

Similarly, $g(\mathcal{A}^t) = \sum_{\tau\in\mathcal{T}^t} R_{f_\tau,m_\tau}\cdot T\cdot\theta_\tau = \sum_{j=1}^{W_t}\rho_j, \forall t = 1, 2, \ldots, t^*$. Denote $\mathcal{G}$ as the solution set obtained by the 2-simple greedy algorithm with initial set $\mathcal{A}^{0*} = \{v_1, v_2\}$, then, based on Eq. (23), we have:

$$
\begin{aligned}
& \frac{g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) + g(\mathcal{G})}{g(\mathcal{A}^*)} \\
&\ge \frac{\sum_{j=1}^{W'}\rho_j + g(\mathcal{G})}{\min_{t=1,2,\ldots,t^*}\{g(\mathcal{A}^t) + W''\theta_{t+1}\} + \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1}} && \text{(25a)} \\
&= \frac{\sum_{j=1}^{W'}\rho_j + g(\mathcal{G})}{\min_{t=1,2,\ldots,t^*}\Big\{\sum_{j=1}^{W_t}\rho_j + W''\rho_{W_t+1}\Big\} + \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1}} && \text{(25b)} \\
&= \frac{\sum_{j=1}^{W'}\rho_j + g(\mathcal{G})}{\min_{k=1,2,\ldots,W'}\Big\{\sum_{j=1}^{k-1}\rho_j + W''\rho_k\Big\} + \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1}} && \text{(25c)}
\end{aligned}
$$

From [44], we have $\sum_{j=1}^{W'}\rho_j / \min_{k=1,2,\ldots,W'}\{\sum_{j=1}^{k-1}\rho_j + W''\rho_k\} \ge 1 - e^{-W'/W''} \ge 1 - 1/e$. On the other hand, based on the maximal marginal benefit criterion of the simple greedy solution set $\mathcal{G}$ and Eq. (18), we have $g(\mathcal{G}) \ge \sum_{v^s_{f,m}\in\bigcup_{i=1}^{N_t}\{v_{t'_i+1}\}} R_{f,m}\cdot T\cdot\theta_{t'_i+1}$. Therefore, we can obtain:

$$
\frac{g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) + g(\mathcal{G})}{g(\mathcal{A}^*)} \ge 1 - 1/e. \qquad \text{(26)}
$$

From the inequality above we have that at least one of the values $g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\})$ and $g(\mathcal{G})$ is greater than or equal to $\frac{1}{2}(1-1/e)\cdot g(\mathcal{A}^*)$. If $g(\mathcal{G}) \ge \frac{1}{2}(1-1/e)\cdot g(\mathcal{A}^*)$, then we have $Q(\mathcal{G}) \ge \frac{1}{2}(1-1/e)\cdot Q(\mathcal{A}^*)$; otherwise, based on the definition of the set function $g(\mathcal{A}) = Q(\mathcal{A}) - Q(\mathcal{A}^{0*})$ and Eq. (21), we have:

$$
\begin{aligned}
Q(\mathcal{A}^{t^*}) &= Q(\mathcal{A}^{0*}) + g(\mathcal{A}^{t^*}) && \text{(27a)} \\
&= Q(\mathcal{A}^{0*}) + g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - [g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - g(\mathcal{A}^{t^*})] && \text{(27b)} \\
&= Q(\mathcal{A}^{0*}) + g(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - [Q(\mathcal{A}^{t^*} \cup \{v_{t^*+1}\}) - Q(\mathcal{A}^{t^*})] && \text{(27c)} \\
&\ge Q(\mathcal{A}^{0*}) + \frac{1}{2}(1-1/e)\cdot g(\mathcal{A}^*) - \frac{1}{2}Q(\mathcal{A}^{0*}) && \text{(27d)} \\
&\ge \frac{1}{2}(1-1/e)\cdot Q(\mathcal{A}^*). && \text{(27e)}
\end{aligned}
$$

Therefore, the larger value of $Q(\mathcal{A}^{t^*})$ and $Q(\mathcal{G})$ is greater than or

equal to $\frac{1}{2}(1-1/e)\cdot Q(\mathcal{A}^*)$, and the theorem is proved.

## REFERENCES


[1] "Global mobile data traffic forecast update, 2015-2020," *Cisco Visual Networking Index (VNI) Forecast*, 2016. [Online]. Available: http://http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] Sandvine, "Global internet phenomena report," 2016. [Online]. Available: https://www.sandvine.com/trends/global-internet-phenomena/

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *submitted to IEEE Communications Surveys and Tutorials, arXiv:1701.01090*, Jan. 2017.

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

[5] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles," in *Proc. ACM Multimedia Systems Conference*, 2011, pp. 133–144.

[6] 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*, 2012, pp. 1107–1115.

[7] J. Sung, M. Kim, K. Lim, and J.-K. K. Rhee, "Efficient cache placement strategy in two-tier wireless content delivery network," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1163–1174, Jun 2016.

[8] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1431–1445, Oct. 2013.

[9] K. Liang, J. Hao, R. Zimmermann, and D. K. Yau, "Integrated prefetching and caching for adaptive video streaming over HTTP: an online approach," in *Proc. ACM Multimedia Systems Conference*, 2015, pp. 142–152.

[10] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[11] F. De Pellegrini, A. Massaro, L. Goratti, and R. El-Azouzi, "Competitive caching of contents in 5G edge cloud networks," *arXiv preprint, arXiv:1612.01593*, 2016.

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

[13] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, "Distributed caching for data dissemination in the downlink of heterogeneous networks," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3553–3568, Oct. 2015.

[14] A. Sengupta, R. Tandon, and O. Simeone, "Cache aided wireless networks: Tradeoffs between storage and latency," in *Proc. Annual Conference on Information Science and Systems (CISS)*, 2016, pp. 320–325.

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

[16] A. Khreishah, J. Chakareski, and A. Gharaibeh, "Joint caching, routing, and channel assignment for collaborative small-cell cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2275–2284, Aug. 2016.

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

[18] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2503–2516, Dec. 2016.

[19] D. H. Lee, C. Dovrolis, and A. C. Begen, "Caching in HTTP adaptive streaming: Friend or foe?" in *Proc. ACM NOSSDAV*, 2014.

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

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

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

[23] W. Li, S. M. Oteafy, and H. S. Hassanein, "Streamcache: popularity-based caching for adaptive streaming over information-centric networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[24] C. Ge, N. Wang, S. Skillman, G. Foster, and Y. Cao, "QoE-driven DASH video caching and adaptation at 5G mobile edge," in *Proc. ACM Conference on Information-Centric Networking*, 2016, pp. 237–242.

[25] C. Li, P. Frossard, H. Xiong, and J. Zou, "Distributed wireless video caching placement for dynamic adaptive streaming," in *Proc. ACM NOSSDAV*, 2016, pp. 1–6.

[26] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2299–2313, Dec. 2007.

[27] Q. Zhang and S. A. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

[28] J. Chen, V. K. Lau, and Y. Cheng, "Distributive network utility maximization over time-varying fading channels," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2395–2404, May 2011.

[29] K. Poularakis and L. Tassiulas, "On the complexity of optimal content placement in hierarchical caching networks," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 2092–2103, May 2016.

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

[31] Z. Li, J. Lin, M.-I. Akodjenou, G. Xie, M. A. Kaafar, Y. Jin, and G. Peng, "Watching videos from everywhere: a study of the PPTV mobile VoD system," in *Proc. ACM Internet Measurement Conference*, 2012, pp. 185–198.

[32] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *Proc. IEEE INFOCOM*, 2014, pp. 82–90.

[33] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[34] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE signal processing magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[35] IBM, "ILOG CPLEX optimization studio." [Online]. Available: http://is.gd/3GGOFp

[36] S. Dash, "Exponential lower bounds on the lengths of some classes of branch-and-cut proofs," *Mathematics of Operations Research*, vol. 30, no. 3, pp. 678–700, 2005.

[37] B. Krishnamoorthy, "Bounds on the size of branch-and-bound proofs for integer knapsacks," *Operations Research Letters*, vol. 36, no. 1, pp. 19–25, 2008.

[38] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[39] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions-II," *Mathematical Programming Studies*, vol. 8, pp. 73–87, 1978.

[40] P. R. Goundan and A. S. Schulz, "Revisiting the greedy approach to submodular set function maximization," *Optimization online*, pp. 1–25, 2007.

[41] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, p. 19, 2012.

[42] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Letters*, vol. 32, no. 1, pp. 41–43, 2004.

[43] "Xiph.org video test media." [Online]. Available: http://media.xiph.org/video/derf/

[44] L. A. Wolsey, "Maximising real-valued submodular functions: Primal and dual heuristics for location problems," *Mathematics of Operations Research*, vol. 7, no. 3, pp. 410–425, 1982.