Optimizing Content Caching and Recommendations with Context Information in Multi-Access Edge Computing

Ana Claudia B. L. Monção, Sand Luz Correa, Aline Carneiro Viana, and Kleber Vieira Cardoso

Abstract—Recently, the coupling between content caching at the wireless network edge and video recommendation systems has shown promising results to optimize the cache hit and improve the user experience. However, the quality of the UE wireless link and the resource capabilities of the UE are aspects that impact user experience and that have been neglected in the literature. In this work, we present a resource-aware optimization model for the joint task of caching and recommending videos to mobile users that maximizes the cache hit ratio and the user QoE (concerning content preferences and video representations) under the constraints of UE capabilities and the availability of network resources by the time of the recommendation. We evaluate our proposed model using a video catalog derived from a real-world video content dataset and real-world video representations and compare the performance with a state-of-the-art caching and recommendation method unaware of computing and network resources. Results show that our approach increases user QoE by at least 68% and effective cache hit ratio by at least 14% in comparison with the other method.

Index Terms—video caching, recommender systems, multi-access edge computing, quality of experience

1 Introduction

According to forecasts presented by Cisco, IP traffic will increase more than three times by 2023, while video will be 82% of all this traffic [1], with a large part of such demand being generated by mobile devices to access video streaming. In this context, it is essential to adopt solutions that take into consideration the quality of experience observed by users and also the availability of resources in the cache, the users' devices, and the network. Recently, the joint use of content caching at the wireless network edge and recommendation systems have exhibited promising results [2], [3].

Content caching at the network edge is not new. Indeed, this idea was introduced by Content Delivery Networks (CDNs) more than 20 years ago. The Multi-access Edge Computing (MEC) paradigm [4], however, introduces some new features to this context. First, MEC provides the opportunity for mobile network providers to offer computing resources at the network edge, which may be also available to service providers. Second, MEC is not limited to offering the storage capacity and basic processing of classic CDNs, it can provide advanced processing, similar to those found in cloud systems, close to end-users [5]. Finally, the MEC platform can provide network status information (e.g., wireless channel conditions, traffic patterns, and user mobility patterns), which can be used to offer better services for users [6]. Thus, MEC is designed not only to allow the execution of traditional services such as video caching, but also new advanced services such as contextualized content, fine-grained recommendation, virtual/augmented/mixed reality, autonomous cars, and industrial automation.

To serve different services and applications, mobile network providers need to deploy or integrate MEC resources in different parts of their networks, such as collocated with the Base Station (BS), collocated with some device of the Radio Access Network (RAN), or collocated with the Core Network (CN) functions (i.e., in the same data center) [5], [7], [8]. Fig. 1 illustrates the last approach (i.e., MEC collocated with the CN functions), which is satisfactory for video streaming services (e.g., Netflix, Hulu, YouTube). In general, the latency from the users until the CN is negligible for video streaming services, and the main bottleneck (i.e., Internet access) can be avoided if the content is cached.

On the other hand, recommendation systems, whose goal is to recommend content that matches the preferences of individual users, have become fundamental components of content delivery services. As an example, Netflix has reported that 80% of the hours streamed by the company comes from recommendations made by its system [9]. This fact illustrates how recommendation systems influence user accesses to content and, thus, they can be employed to induce access patterns that improve the cache performance. In [3], the authors elaborate on this idea and present an optimization model, denoted Joint Caching and Recommendation Problem (JCRP). JCRP combines caching and video content recommendations to maximize the cache hit ratio, while minimally affecting user preferences. The recommendation system is used to shape the user demand towards content that can be shared by multiple users, resulting in an improved cache hit ratio and also user quality of experience (QoE).

Nevertheless, caching and recommending popular content at the network edge is not enough to ensure the user QoE. Context factors such as the capabilities of the user

- *Ana Claudia B. L. Monção, Sand Luz Correa and Kleber Vieira Cardoso are with Instituto de Informática, Universidade Federal de Goiás, Goiânia, Brasil.*
  E-mail:anaclaudia@inf.ufg.br, sand@inf.ufg.br, kleber@inf.ufg.br
- *Aline Carneiro Viana is with Inria, France. E-mail:aline.viana@inria.fr*

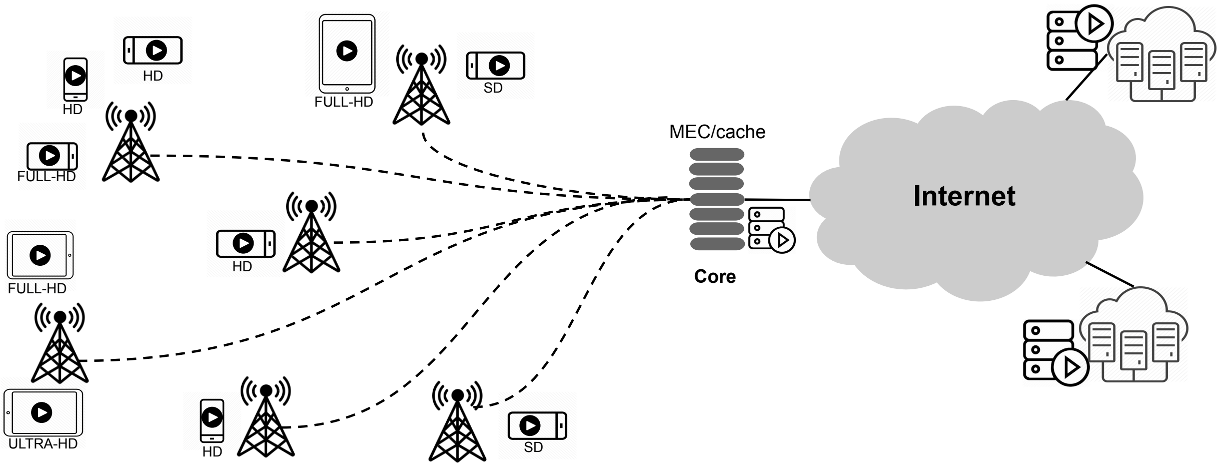


Fig. 1. Reference scenario for the system model.

equipment (UE) to reproduce the video and the quality of its wireless link may also affect the user satisfaction [10]. Thus, videos with resolutions that the user mobile device (UE) cannot reproduce should not be among her recommendations, even if the content is of her interest. Similarly, videos encoded with representations that require more throughput than the present user wireless link capacity should not be included in the user recommendations. Therefore, information on the UE capabilities and the quality of the wireless link, by the time of the video recommendation, is important to provide a better service to users. Despite these facts, as far as we know, no previous work has tackled the problem of caching and recommending videos in mobile networks taking into account content preferences and the availability of the computing resources in the UE and the quality of its wireless link.

In this work we propose an optimization model, denoted *Resource-Aware Video Recommendation* (RAViR), for the joint task of caching and recommending videos to mobile users, taking into consideration content preferences and information on the availability of computing and network resources (provided by the MEC platform) by the time of the video recommendation. Consider, for example, a streaming service composed of a video cache service deployed in the CN and serving mobile users with different UE capabilities and under different wireless link conditions, as illustrated in Fig. 1. The general idea is that the recommendation system of the streaming service should recommend to individual users videos that i) match properly their content preferences; ii) provide the best QoE according to the UE capabilities and the quality of the wireless link by the time of the recommendation; and iii) have the potential to be demanded from multiple users.

This paper advances the state-of-the-art in the interplay between caching and recommendation systems by making the following contributions:

- We formulate and solve an optimization problem for the joint task of caching and recommending videos to mobile users, considering content preferences and the availability of computing and network resources. The objective of RAViR is to recommend videos to users that maximize the cache hit ratio and the user QoE (concerning content preferences and video representations) under the constraints of UE capabilities and the availability of network resources by the time of the recommendation.
- We provide an extensive evaluation of RAViR using a real dataset from the MovieLens project [11] and compare the obtained results with different variants of the JCRP [3]. Results show that RAViR increases user QoE by at least 68% and effective cache hit ratio by at least 14% in comparison with the JCRP variants.
- We introduce a new metric to assess the QoE of mobile users that consume streaming services in the era of MEC. Our QoE metric is designed to capture the capabilities of the UE as well as the effect of the availability of computing and network resources on the user experience.

The remainder of this paper is organized as follows. In Section 2, we discuss the related work. The system model and the problem formulation are presented in Sections 3 and 4, respectively. We evaluate the performance of RAViR in Section 5. Section 6 concludes this paper and presents future directions.

2 Related Work

Caching is a classical subject in computer science and has been well studied in computer networks [12], [13]. However, the expected wide adoption of MEC in 5G networks is motivating the research community to revisit the topic under a refreshed perspective. This section briefly presents an overview of related research on caching and recommendation systems as well as QoE in video streaming services in the context of 5G networks.

2.1 Caching and Recommendation Systems

Few works in the literature have based caching decisions on personalized recommendations issued by recommendation systems [3], [14], [15], [16], [17]. From this group, the first four deals with video content, and only [16] and [3] tackled the problem in the context of the network edge. The authors in [16] introduce the concept of “soft cache hits” in the context of entertainment-oriented content consumption on the

Internet. The idea is to recommend similar cached content to mobile users that request videos which are not cached at the base stations. Since the majority of video content on the Internet is entertainment-oriented, the authors argue that users are likely to accept the recommendations.

The authors of [3] evolve the initial idea introduced in [16] and propose a model to tackle the Joint Caching and Recommendation Problem (JCRP) in small cell networks. In JCRP, the recommendation system is used not only as a predictor of video content demand but also as a demand-shaping tool to enhance cache performance. To this end, the JCRP model issues recommendations that may not necessarily rank top in the inferred user content preferences but still score high in them. By bounding the distortion that such recommendations introduce to the original user content preferences, the JCRP model controllably guides user demand toward content that attracts the preference from multiple users. The authors show that using recommendation systems as a traffic engineering tool brings relevant gains in cache performance without significantly degrading the user preferences. However, in [3] user preferences are defined only in terms of content. Video resolution and the computing and network resources available at the user side to play the video are not take into consideration. As a result, JCRP may recommend videos which representations are not suitable for the available user resources. In addition, although the cache is assumed to be collocated with multiple BSs, the authors do not describe how this distributed cache is managed, which is a non-trivial problem. Similar to [3], in our work, we also use recommendation systems as a traffic engineering mechanism to enhance the cache hit. However, different from [3], in our work, user preferences comprise video content explicitly and video representation implicitly. The latter is obtained from UE capabilities and from the prediction on the network resources available to the UE, which is provided by the MEC platform by the time of the recommendation. Also, different from [3], we assume that the cache is collocated with the CN.

2.2 QoE in Video Streaming Services

The work in [3] assesses the user QoE using only the cache hit ratio. However, the cache hit ratio is not enough to measure if the video content is being delivered to the users with the appropriate quality of service (QoS). Indeed, many papers have investigated how to measure or predict the QoE of mobile users when they consume video streaming services. For example, a work from Netflix [18] proposes different ways to assess the user QoE by combining images and mobility metrics. Most of these metrics compare different video encodings with evaluations issued by the users. In [19], the authors propose a low-complexity network metric to derive the QoE perceived by users of UHD video flows in 5G networks. This metric, denoted Congestion Index (CI), represents the ability of the network to successfully deliver video stream based on the maximum available bandwidth on the path from the server to the user. The QoE is then obtained using a regression technique that correlates the CIs (resulting from variable video bitrates and available bandwidth) with subjective user assessments. Indeed, most of the QoE metrics for video streaming services proposed in the literature are based solely on video transmission parameters (e.g., video bitrates and network throughput). Differently, in our work, we first propose a new Congestion Index (CI) that assesses the ability of the network to deliver video stream when the video comes from the edge cache or the remote cloud. In the last case, the CI is higher since the UE faces competition for the bottleneck link to access the Internet. We then define a new QoE metric that accounts for the effect of the UE capabilities, the CI, and over/underestimations of the network conditions on the user quality of experience.

3 System Model

In this work, we consider a mobile network with a MEC system as illustrated in Fig. 1. Multiple BSs, in different locations, offer connectivity to a set of mobile users, denoted by $\mathcal{U}$. These users consume video streaming services from a video catalog, represented by $\mathcal{I}$. Since video streaming services are adequately provisioned by cache stored in the core network, in this work, we assume that a cache service is deployed in MEC hosts at the CN. The cache service has a limited total storage capacity, represented by C, which is measured in normalized file size units. At a given instant of time, the cache stores a subset $\mathcal{I}'$ of the entire catalog. Servers in remote data centers host copies of the entire catalog.

Let $\mathcal{V}$ denote the set of video content (e.g., The Godfather, The Godfather: Part II, Schindler's List, Pulp Fiction, etc.) available in the catalog $\mathcal{I}$ and let $\mathcal{R}$ be the set of all possible video representations, where a video representation $r \in \mathcal{R}$ is composed of bitrate and resolution, i.e., ($Btr(r)$,$Res(r)$). Each video content $v \in \mathcal{V}$ is available into a subset $\mathcal{R}_v \subseteq \mathcal{R}$ of representations. We use the notation v_r to denote a video content $v \in \mathcal{V}$ encoded into a representation $r \in R_v$, so that v_r has an encoding bitrate $Btr(r)$, a corresponding video resolution $Res(r)$, and a size $Size(v_r)$. Thus, the catalog $\mathcal{I}$ is composed of video content with their representations, i.e., $\mathcal{I} = \{v_r | v \in \mathcal{V}, r \in \mathcal{R}_v\}$.

Next, we describe how we capture user preferences on content, computing and network resources, and the impact of recommendations on user choices in our model. Table 1 summarizes the notation used in this work.

3.1 Users and Computing and Network Resources

We assume that each user $u \in \mathcal{U}$ consumes video streaming services through a mobile device (UE) connected to a base station. Each mobile device presents different capabilities (e.g., processor type, graphic card, memory, and screen size) and, consequently, may support a different maximum resolution for video playback, denoted by $ResUE(u)$.

At any point in time, each user $u \in \mathcal{U}$ receives a video streaming through her wireless link. The quality of the UE wireless link may vary significantly over time due to several factors, including mobility and signal impairments. We consider that the quality of the wireless link can be periodically obtained by the video streaming service through the MEC Radio Network Information Service (RNIS) [6]. This quality is reported as the Channel Quality Indicator (CQI), one of the key parameters in the Channel State Information (CSI) [20]. The base station uses the CQI parameter to

TABLE 1
Notation Table

Notation	Description
C	Total cache storage capacity
N	Number of recommendations per user
$\mathcal{U}$	Set of users
$\mathcal{V}$	Set of video content
$\mathcal{R}$	Set of representations (bitrate, resolution)
$\mathcal{R}_v$	Set of representations for a video content v
$\mathcal{I}$	Set of all video content with their representations
$\mathcal{I}'$	Cache content
$\mathcal{T}$	Set of video thematic categories
$\mathcal{Q}$	Set of possible Channel Quality Indicator (CQI) values
v_r	Video content v encoded in representation r
$Size(v_r)$	Size of video content v_r
$Btr(r)$	Encoding bitrate of video representation r
$Res(r)$	Video resolution of video representation r
$ResUE(u)$	Maximum resolution supported by the UE of user u
$CQI(u)$	Channel Quality Indicator of the UE wireless link of user u
$BtrAvl(q)$	Bitrate associated with CQI value q
$ResCQI(q)$	Video representation suitable to the CQI value q
$\mathbf{f}^v$	Thematic categories vector of the video content v
$\mathbf{f}^u$	Thematic preferences vector of the user u
$Sim(u, v)$	Similarity between user preference u and video content v
P_u^{Cont}	Video content preference distribution of user u
P_u^{Res}	Video representation preference distribution of user u
P_u^{Pref}	Preference distribution of user u
P_u^{Rec}	Probability distribution due to recommendation
P_u^{Req}	Content item request probability distribution of user u (recommended items)
$P_u^{Req\sim}$	Content item request probability distribution of user u (non-recommended items)
$Best(u)$	The most suitable video representation to user u
$Sat(u, r)$	Satisfaction of user u with a video representation r
w_u	Weight that user u gives for a recommendation
W_u	Recommendation window of user u
K_u	Size of recommendation window W_u
$Tol(u)$	Distortion tolerance on recommendation accepted by user u

determine the appropriate Modulation and Coding Scheme (MCS) between itself and the UE. The CQI also influences the transport block size (TBS), which is the size of a data unit, i.e., a transport block from the MAC layer given to the PHY layer. With the MCS and TBS information, the UE wireless link capacity (i.e., the bitrate) can be determined.

Let denote by $\mathcal{Q}$ the set of all possible CQI values. Each $q \in \mathcal{Q}$ is associated with a given pair MCS and TBS. Thus, it is possible to determine the UE wireless link capacity (in bitrate) for each $q \in \mathcal{Q}$ [21]. We represent such UE wireless link capacity by $BtrAvl(q)$. Also, we denote by $CQI(u) \in \mathcal{Q}$ the predicted average quality of the UE wireless link of user u by the time of the recommendation.

3.2 Content Preferences

We assume that a video content is categorized by one or more thematic categories (e.g., Animation, Adventure, Romance, Comedy, etc.). Formally, let $\mathcal{T}$ be the set of all possible thematic categories. Each video content $v \in \mathcal{V}$ is associated with a feature vector $\mathbf{f}^v$ whose j-th element $\mathbf{f}^v(j)$, $j \in \mathcal{T}$, represents the adherence level of category j to video content v. This adherence level is normalized and assumes values in the range [0,1], so that $\sum_{j \in \mathcal{T}} \mathbf{f}^v(j) = 1, \forall v \in \mathcal{V}$.

Similarly, each user $u \in \mathcal{U}$ is associated with a feature vector $\mathbf{f}^u$, where each element $\mathbf{f}^u(j)$, $j \in \mathcal{T}$, represents the interest of user u in thematic category j. This interest is estimated based on the video content watched and evaluated by the user, and is also normalized so that $\sum_{j \in \mathcal{T}} \mathbf{f}^u(j) = 1, \forall u \in \mathcal{U}$.

Given the feature vectors $\mathbf{f}^v$ and $\mathbf{f}^u$, we can estimate the interest of user u in the video content v, denoted by $Sim(u, v)$, using the cosine similarity [3], i.e.,

$$Sim(u, v) = \frac{\sum_{j \in \mathcal{T}} \mathbf{f}^u(j) \cdot \mathbf{f}^v(j)}{\sqrt{\sum_{j \in \mathcal{T}} \mathbf{f}^u(j)} \cdot \sqrt{\sum_{j \in \mathcal{T}} \mathbf{f}^v(j)}}. \quad (1)$$

Thus, for each user $u \in \mathcal{U}$ and each video content $v \in \mathcal{V}$ we can compute the probability of user u be interested in the video content v as:

$$P_u^{Cont}(v) = \frac{Sim(u, v)}{\sum_{z_s \in \mathcal{I}} Sim(u, z)}, \quad (2)$$

where $z \in \mathcal{V}$ and $s \in \mathcal{R}$ are, respectively, the content and representation associated with video $z_s \in \mathcal{I}$, and $\sum_{v_r \in \mathcal{I}} P_u^{Cont}(v) = 1$. Indeed, $P_u^{Cont}(v), \forall v_r \in \mathcal{I}$ represents the content preference distribution of user u over the catalog $\mathcal{I}$. We denote this distribution by P_u^{Cont}.

The demand for video content is time-varying. Usually, it grows for some finite time after the video content becomes available in the catalog, and then it gradually fades out [22], [23]. Similar to other works [3], [24], [25], in this paper, we assume that users' content request patterns change slowly over time, usually in the order of a few hours within a day. Thus, content demand predictions, recommendations, and caching decisions are made at this time scale.

3.3 Representation Preferences

In addition to content interest, the quality in which a video is delivered affects the user experience positively or negatively. Thus, videos with resolutions that the user mobile device can not reproduce or videos encoded with representations that require more capacity than the one available in the UE wireless link should not be among her preferences, even with interesting content.

Formally, let $CQI(u) \in \mathcal{Q}$ be the predicted average quality of the UE wireless link by the time the system has to issue a recommendation to user u. Since the bitrate $BtrAvl(CQI(u))$ is known for each $q \in \mathcal{Q}$, we can compute the most suitable video representation $ResCQI(CQI(u))$ that can be transmitted under this wireless link condition. The best video representation for user u by the time of the recommendation is given by:

$$Best(u) = \min(ResUE(u), ResCQI(CQI(u))). \quad (3)$$

To capture the relevance of a given video representation $r \in \mathcal{R}$ to user u by the time of the recommendation, we define a representation relevance factor, denoted by $Sat(u, r)$, whose values varies in the range [0,1] and is given by:

$$Sat(u,r) = \begin{cases} 0, & Res(r) > Best(u) \\ 1, & Res(r) = Best(u) \\ 1 - \frac{(Best(u) - Res(r))}{Best(u)}, & Res(r) < Best(u). \end{cases} \tag{4}$$

The rationale behind Equation (4) is that the most appropriate video representation for user u, given $ResUE(u)$ and $ResCQI(CQI(u))$, achieves the relevance factor of 1, while the other representations show a progressive reduction in the user interest. Representations that are not appropriate to the user mobile device will score 0.

By applying the representation relevance factor $Sat(u, r)$ to each video representation r available in the catalog $\mathcal{I}$, we obtain a new distribution, denoted by P_u^{Res}, that describes the user preferences on video representations by the time of the recommendation, i.e.,

$$P_u^{Res}(r) = \frac{Sat(u,r)}{\sum_{z_s \in \mathcal{I}} Sat(u,s)}, \tag{5}$$

where $z \in \mathcal{V}$ and $s \in \mathcal{R}$ are, respectively, the content and representation associated with video $z_s \in \mathcal{I}$, and $\sum_{v_r \in \mathcal{I}} P_u^{Res}(r) = 1$. Thus, on the user side, we distinguish between video content preferences (P_u^{Cont}) and video representation preferences (P_u^{Res}). We then define a new distribution, called user preferences (P_u^{Pref}) as follows.

Definition 3.1. The user preference distribution, denoted by P_u^{Pref}, is a combination of the user content preference distribution (P_u^{Cont}) and the video representation preferences distribution (P_u^{Res}) and captures the joint preferences of user u on a video content $v \in \mathcal{V}$ and a video representation $r \in R_v$ by the time of the recommendation, for all items in the catalog, i.e.,

$$P_u^{Pref}(v_r) = \frac{P_u^{Cont}(v) \cdot P_u^{Res}(r)}{\sum_{z_s \in \mathcal{I}} P_u^{Cont}(z) \cdot P_u^{Res}(s)}, \tag{6}$$

where $z \in \mathcal{V}$ and $s \in \mathcal{R}$ are, respectively, the content and representation associated with video $z_s \in \mathcal{I}$, and $\sum_{v_r \in \mathcal{I}} P_u^{Pref}(v_r) = 1$. Equation (6) comprises the preferences of user u in terms of video content and video representation. The latter, on the other hand, depends on computing and network resources, i.e., the user mobile device capabilities and the UE wireless link condition by the time of the recommendation. Thus, the higher the value of $P_u^{Pref}(v_r)$, the better the user experience.

3.4 Impact of Recommendations

The literature shows that recommendation systems have a high impact on user choices. In general, recommendation systems increase requests for recommended items while proportionally decrease the demand for other items in the catalog [26]. There is also strong evidence that the number of recommended items and their respective positions on the recommendation list influence user choices [14]. Thus, the content that users eventually request also depend on the recommendations issued to them.

To capture the impact of recommendations on user choices, we assume a new probability distribution, denoted by P_u^{Rec}, over the set of video content ($\mathcal{V}$). Similar to [3], we assume that this distribution equally boosts all items in the recommendation list, as defined in the following.

Definition 3.2. Let $\mathcal{V}_u \subseteq \mathcal{V}$ be the set of content recommended to user u, where $\mathcal{V}_u$ is composed of the top N ($N = |\mathcal{V}_u|$) content in u's content preference distribution P_u^{Cont}. The probability distribution due to recommendations P_u^{Rec} is given by:

$$P_u^{Rec}(v) = 1/N, \quad \forall v \in \mathcal{V}_u. \tag{7}$$

The intuition behind Equation (7) is that the shorter the recommendation list, the greater its influence on the user choice. This is especially true for mobile users, where devices with small screens limit the amount information exhibited at once. Thus, recommendation systems modulate the original user video preferences P_u^{Pref} to yield the ultimate video request distribution of user u. The latter is defined as follows.

Definition 3.3. Let w_u be the weight that expresses the importance user u gives to recommendations. The ultimate video request distribution of user u over the catalog $\mathcal{I}$, denoted by P_u^{Req}, is a function of the user preference distribution by the time of the recommendation (P_u^{Pref}) and the probability distribution due to recommendations (P_u^{Rec}) and is given by

$$P_u^{Req}(v_r) = w_u \cdot P_u^{Rec}(v) + (1 - w_u) \cdot P_u^{Pref}(v_r), \tag{8}$$

for the videos that are recommended to user u, i.e., $\forall v_r \in \mathcal{I} \mid v \in \mathcal{V}_u$, and by

$$P_u^{Req\sim}(v_r) = (1 - w_u) \cdot P_u^{Pref}(v_r), \tag{9}$$

for the videos that are not recommended to user u, i.e., $\forall v_r \in \mathcal{I} \mid v \notin \mathcal{V}_u$.

3.5 Recommendation Systems as a Traffic Engineering Tool

The authors in [3] use recommendation systems to modulate user preferences and, at the same time, optimize content cache policies. To this end, instead of issuing recommendations for the top N items on the user preference distribution P_u^{Pref}, as usually expected for a recommendation system, the system selects N items among the ones residing within a recommendation window W_u. This window is defined by the top K_u items of P_u^{Pref}, where $K_u > N$. Indeed, K_u is defined according to a distortion tolerance $Tol(u) \in [0, 1)$, which controllably bounds the distortion that recommendations introduce to the original user preferences.

Assuming a list W_u, $W_u \subseteq \mathcal{I}$, with K_u items, the last N items will be recommended in the worst case, i.e., those ranked in positions $K_u - N + 1, K_u - N + 2, .., K_u$. Denoting by i an item in W_u and by $pos_u(i)$ the position of i in W_u, the worst case distortion is defined as:

$$\Delta_u(K_u, N) = 1 - \frac{\sum_{i: pos_u(i) \in [K_u - N + 1, K_u]} P_u^{Pref}(i)}{\sum_{i: pos_u(i) \in [1, N]} P_u^{Pref}(i)}. \tag{10}$$

The cardinality K_u of the expanded recommendation window W_u is given by:

$$K_u = max\{k|\Delta_u(K_u, N) \leq Tol(u)\}. \tag{11}$$

In this work, we use the approach proposed in [3] to modulate user preferences and to optimize content cache policies. However, the definition of preferences in [3] does not account for video representations, and each item $i \in \mathcal{I}$ is only described by its content (i.e., $i = v$). As a consequence, $P_u^{Pref} = P_u^{Cont}$. Different from [3], the definition of preferences in our work takes into account video content (v) and representation (r). Thus, each item $i \in \mathcal{I}$ is a video content encoded into a certain representation (i.e., $i = v_r$) and user preferences are defined in terms of both content and representation, as stated in Equation (6).

4 Cache Policy and Resource-aware Recommendation System

In this section, we define a new joint cache and recommendation problem, called *Resource-Aware Video Recommendation* (RAViR), which takes into consideration content preferences, information on the availability of computing and network resources by the time of the video recommendation, and the cache hit ratio.

Consider a cache service running in MEC hosts at the CN with total storage capacity C; the catalog $\mathcal{I}$ composed of video content, their representations, and their lengths $Size(v_r)$, $v_r \in \mathcal{I}$; the set $\mathcal{U}$ of users with their distributions of preferences (P_u^{Pref}, $u \in \mathcal{U}$), recommendation (P_u^{Rec}, $u \in \mathcal{U}$) and request (P_u^{Req}, $u \in \mathcal{U}$) given by Equations (6), (7), and (8) and (9), respectively; and user recommendation windows $W_u \subseteq \mathcal{I}$ with K_u videos, where K_u is given by Equation (11) and $u \in \mathcal{U}$. The goal of RAViR is to recommend, for every user $u \in \mathcal{U}$, N videos from the recommendation window W_u, so that the recommended videos maximize the cache hit ratio and the user experience.

Formally, let $\{y_{v_r}\}$, $v_r \in \mathcal{I}$, be a set of binary decision variables, so that $y_{v_r} = 1$ if video v_r is in cache and $y_{v_r} = 0$, otherwise. Let $\{x_{u,v_r}\}$, $u \in \mathcal{U}, v_r \in \mathcal{I}$, be another set of binary decision variables, so that $x_{u,v_r} = 1$ if v_r is recommended to user u and $x_{u,v_r} = 0$, otherwise. We formulate the RAViR problem as follows:

$$\max_{y,x} \sum_{u \in \mathcal{U}} \sum_{v_r \in W_u} y_{v_r}(x_{u,v_r} \cdot P_u^{Req}(v_r) + (1 - x_{u,v_r}) \cdot P_u^{Req\sim}(v_r)) \tag{12}$$

subject to:

$$\sum_{v_r \in \mathcal{I}} y_{v_r} \cdot Size(v_r) \leq C \tag{13}$$

$$\sum_{v_r \in W_u} x_{u,v_r} = N, \forall u \in \mathcal{U} \tag{14}$$

$$P_u^{Res}(v_r) > 0, \forall u \in \mathcal{U}, \forall v_r \in W_u \tag{15}$$

$$y_{v_r}, x_{u,v_r} \in \{0, 1\}, \forall u \in \mathcal{U}, \forall v_r \in W_u. \tag{16}$$

Equation (12) maximizes the cache hit ratio and the user experience using each user preference list W_u. This list comprises preferences on content (explicitly) and resolution (implicitly). Equation (13) reflects the cache storage capacity constraint, while Equation (14) ensures that exactly N videos are recommended to every user. Finally, Equation (15) ensures that the recommended video does not exceed the computing and network resources to every user.

5 Experimental Evaluation

This section evaluates the performance of RAViR using a video catalog derived from real-world data. In the following, we first describe the methodology we use to obtain the video catalog (Section 5.1). Then, we i) detail how we obtain information on computing and network resources related to the user mobile devices (Section 5.2), ii) present the baseline algorithms used in the evaluation (Section 5.3), and iii) introduce the metrics used to assess the performance of the evaluated methods (Section 5.4). Finally, we discuss the obtained results (Sections 5.5 and 5.6).

5.1 Dataset and Pre-processing

In order to create the video catalog, we use the MovieLens project dataset [11] to obtain video content and user content preferences. The MovieLens dataset is a collection of 5-star movie ratings collected from an online movie recommendation service called MovieLens. This dataset has been used in related work on caching and recommendations [3], [27] and consists of approximately 100,000 user ratings (ranging from 0 to 5) applied to a content catalog of 9,742 movies ($|\mathcal{V}| = 9,742$), characterized by 20 thematic categories ($|\mathcal{T}| = 20$), and a community of 610 users ($|\mathcal{U}| = 610$).

Since the MovieLens dataset does not provide different video representations for content, we use video qualities supported by YouTube to derive a set of video representations. More specifically, we consider 6 video resolutions, and their corresponding YouTube recommended bitrates [21], presented in Table 2, to compose the set of video representations ($|\mathcal{R}| = 6$).

TABLE 2
Video representations used in our evaluation

Type $Type(r)$	Name	Video Representation $(Btr(r), Res(r))$
1	medium	(1.5Mbps,360px)
2	large	(4Mbps,480px)
3	hd720	(7.5Mbps,720px)
4	hd1080	(12Mbps,1080px)
5	hd1440	(24Mbps,1440px)
6	hd2160	(53Mbps,2160px)

We then combine this video representation set ($\mathcal{R}$) and the MovieLens dataset ($\mathcal{V}$) to generate the final video catalog $\mathcal{I}$. For each video content $v \in \mathcal{V}$, we generate a number n_v that indicates the amount of representations of v available in the catalog $\mathcal{I}$, with n_v sampled from a Uniform distribution $U(1, 6)$. Given v and n_v, we insert in the catalog $\mathcal{I}$ all videos v_r with content v encoded in representation r, so that $Type(r) \in \{1, \ldots, n_v\}$ and $Type(r)$ denotes the type associated to representation r (Table 2). This approach ensures that every video content $v \in \mathcal{V}$ is available in a subset of representations $\mathcal{R}_v$ with increasing video qualities, starting from the lowest one. The size of the video ($Size(v_r)$, $v_r \in \mathcal{I}$) is a function of its duration and representation, employing a normalized scale compatible with the cache storage. After combining the MovieLens dataset ($\mathcal{V}$) and the set of representations (R), we end up with a video catalog of approximately 35,000 items ($|\mathcal{I}| = 35,000$).

We also use the MovieLens dataset ($\mathcal{V}$) to compute the feature vectors $\mathbf{f}^v$ and $\mathbf{f}^u$ and then infer the user content preference P_u^{Cont}. To this end, we use the same approach employed in [3] and rely on the information that a user rates a specific content, rather than the actual rating she assigned to it. More specifically, if a content $v \in \mathcal{V}$ is classified into t categories, then the positions corresponding to such categories in the video feature vector $\mathbf{f}^v$ are set to $1/t$ and the remaining ones to zero. To illustrate, let suppose that content v is described by categories "Action" and "Comedy". We set $\mathbf{f}^v(j) = 0.5$ for those j values corresponding to categories "Action" and "Comedy", and $\mathbf{f}^v(j) = 0$ to the other categories. Denote by $\mathcal{A}_u \subseteq \mathcal{V}$ the set of video content rated by user $u \in \mathcal{U}$. We estimate the element $\mathbf{f}^u(j)$, $j \in \mathcal{T}$, of user feature vector $\mathbf{f}^u$ as:

$$\mathbf{f}^u(j) = \frac{\sum_{v \in \mathcal{A}_u} \mathbf{f}^v(j)}{\sum_{j \in \mathcal{T}} \sum_{v \in \mathcal{A}_u} \mathbf{f}^v(j)}. \tag{17}$$

Once the feature vectors $\mathbf{f}^v$ and $\mathbf{f}^u$ are computed for every $v \in \mathcal{V}$ and for every $u \in \mathcal{U}$, respectively, we use Equations (1) and (2) to estimate P_u^{Cont} for every $u \in \mathcal{U}$.

5.2 Computing and Network Resources and Ultimate User Request

As part of the input to our model, we need to provide computing capabilities related to the user mobile device and the prediction of the average quality of the UE wireless link by the time of the recommendation. To represent the UE computing capabilities, we assign the maximum supported resolution ($ResUE(u)$, $u \in \mathcal{U}$) by taking a random value from a Uniform distribution $U(1,6)$. This ensures that $ResUE(u) = Res(r)$ for some video representation r described in Table 2. Similar to the previous random choices employed, this simple approach has the sole purpose of exercise the model without bias. Any statistical knowledge about UE computing capabilities or video representations could influence the specific values observed in the evaluation, but not the general trends.

For the quality of the UE wireless link ($CQI(u) \in \mathcal{Q}$, $u \in \mathcal{U}$), values of signal strength-related metrics (e.g., the Arbitrary Strength Unit - ASU) can be grouped into CQI values, e.g., $|\mathcal{Q}| = 15$, as illustrated in Table 3 (column 1) [21]. Given that each CQI value $q \in \mathcal{Q}$ is associated with an MCS and a TBS, there is a mapping from CQI values to bitrates, also illustrated in Table 3 (column 4). Combining the information presented in Tables 2 and 3, it is possible to determine the best video representation for a certain CQI value. This information is shown on the right side of Table 3 (column 5). Again, in order to avoid bias, we obtain the predicted average quality of the UE wireless link ($CQI(u)$) by sampling from a Uniform distribution $U(1,15)$ (i.e., the interval presented in column 1 of Table 3).

Having $ResUE(u)$ and $CQI(u)$, $u \in \mathcal{U}$, and using Equation (3), we can compute the best representation for user u by the time of the recommendation. Also, using Equations (4), (5), and (6), we can estimate P_u^{Pref}, $\forall u \in \mathcal{U}$. We compute P_u^{Rec}, $\forall u \in \mathcal{U}$, by applying Equation (7) for the top N video content in P_u^{Cont}. In our evaluation, we experiment with $N = 3$ and $N = 5$. Given P_u^{Rec} and P_u^{Pref}, $\forall u \in \mathcal{U}$, we compute the ultimate video request distribution (P_u^{Req} and $P_u^{Req\sim}$, $\forall u \in \mathcal{U}$) using Equations (8) and (9). Aligned with [3], we sample the user recommendation weight (w_u, $\forall u \in \mathcal{U}$), from a Uniform distribution $U(0.5, 0.7)$.

TABLE 3
Mapping between CQI values, bitrates, and video representations

CQI	**MCS**	**TBS**	**Bitrate**	**Video Representation**
q			$BtrAvl(q)$	$ResCqi(q)$
Value	Index	(bits)	(Mbps)	(Mbps,px)
1	0	1384	2.768	(1.5Mbps,360px)
2	0	1384	2.768	(1.5Mbps,360px)
3	2	2216	4.432	(4Mbps,480px)
4	4	3624	7.548	(7.5Mbps,720px)
5	6	5160	10.320	(7.5Mbps,720px)
6	8	6968	13.936	(12Mbps,1080px)
7	11	8760	17.520	(12Mbps,1080px)
8	13	11448	22.896	(12Mbps,1080px)
9	16	15264	30.528	(24Mbps,1440px)
10	18	16416	32.832	(24Mbps,1440px)
11	21	21384	42.768	(24Mbps,1440px)
12	23	25456	50.912	(24Mbps,1440px)
13	25	28336	56.672	(53Mbps,2160px)
14	27	31704	63.408	(53Mbps,2160px)
15	27	31704	63.408	(53Mbps,2160px)

5.3 Baseline Methods

We compare RAViR against JCRP. However, user preferences in JCRP are based on video content and do not account for video representation. Thus, to provide a fair comparison between the two methods, we adapt the JCRP model to limit the video representations considered during the generation of the user recommendation window W_u. This adaptation does not affect the output issued by the method since, for JCRP, videos with different representations, but the same content, have the same value for the user. This approach leads us to three variants of the original method:

- JCRP-L: given a set of videos with the same content, this variant selects the representation with the lowest resolution to be included in W_u.
- JCRP-M: given a set of videos with the same content, this variant selects the representation with medium resolution to be included in W_u.
- JCRP-H: given a set of videos with the same content, this variant selects the representation with the highest resolution to be included in W_u.

5.4 Evaluation Metrics

Our evaluation is focused on two metrics: the user QoE and the cache hit ratio (CHR). In the following, we describe these metrics in detail.

The QoE metric estimates the user's satisfaction when watching a video. In our work, this metric is computed for each video recommended to the user. The QoE is high when the recommended video is in cache and its representation matches the computing and network resources available to the UE. More specifically, the following factors affect the user QoE negatively:

- **F1**: The recommended video is not in cache. In this case, the video will be delivered by the cloud server, and the UE has to face competition for the bottleneck link to access the Internet.
- **F2**: The recommended video requires a higher bitrate than that available on the UE wireless link. In this case, the video reproduction is subjected to delays and stalls.
- **F3**: The recommended video requires a higher resolution than the one supported by the user mobile device. In this case, the UE is not be able to reproduce the video.
- **F4**: The recommended video requires a lower resolution than that supported by the available resources (UE computing capabilities and wireless link). In this case, the available resources are not being used properly and the user is receiving a video with lower quality than her device is able to present.

In [19], the authors propose a correlation between the user QoE and the Congestion Index (CI), a metric that represents the ability of the network to successfully deliver a video stream based on the maximum available bandwidth on the path from the server to the user. This correlation is shown in Equation (18):

$$QoE = \frac{5.082}{\sqrt{CI}} - 0.891, \quad CI > 0, \tag{18}$$

where CI is the ratio of the maximum required bandwidth for the stream divided by the available network bandwidth. The authors in [19] obtain this equation using a regression technique over subjective user assessments.

In our work, the recommended video can be delivered by the MEC hosts or by the cloud servers, depending on whether the video is in the cache or not. To capture this situation, we define an Internet Congestion Index metric, denoted by ICI, that represents the current state of the network path between the MEC hosts and the cloud servers. If the video is in the cache, no access to the cloud servers is needed, and $ICI = 0$. Otherwise, $ICI > 0$ and its value will depend on the utilization of the links connecting the two storage places (i.e., MEC and cloud). We then propose a new CI ratio to account for the ICI ratio as follows. Given a video $v_r \in \mathcal{I}$ recommended to user $u \in \mathcal{U}$, we compute the CI for such recommendation as:

$$CI(u, r) = \frac{Btr(r)}{BtrAvl(CQI(u))} * (1 + ICI), \tag{19}$$

where $Btr(r)$ and $BtrAvl(CQI(u))$ represent, respectively, the maximum required bitrate for the stream and the bitrate available in the UE wireless link.

Equations (18) and (19) work well to capture the user QoE concerning video transmission, e.g., the situations described in **F1** and **F2**. However, it fails to capture the QoE in other situations. For instance, for situation **F4**, these equations return a high user QoE, when it is expected otherwise. To capture the QoE of user $u \in \mathcal{U}$ when receiving a recommended video $v_r \in \mathcal{I}$ and taking into account all situations described from **F1** to **F4**, we propose a new QoE metric, based on Equation (18), as follows:

$$QoE(u, v_r) = \begin{cases} 0, & r > Best(u) \\ \frac{5.082}{\sqrt{CI(u,r)}} - 0.891, & r = Best(u) \\ \frac{5.082}{\sqrt{1/CI(u,r)}} - 0.891, & r < Best(u), \end{cases} \tag{20}$$

where CI is given by Equation (19). The first part of Equation (20) captures the user QoE in case of **F3** happens; the second part of the equation applies for **F1** and **F2**. Finally, the third part deals with **F4**.

For the cache hit ratio (CHR), we adapt the metric employed in [3], which is based on the probability of cached videos being requested. In [3] and in most literature on video caching, a recommendation is considered a hit if the video content is in the cache. These works either assume that all representations (of the content) are in the cache, or that the highest representation is cached and transcoding techniques will be in charge of transforming such representation to the one expected by the user. Both strategies, however, rapidly consume the available cache storage. Indeed, Netflix has reported achieving the same caching efficiency with 50% less storage by caching videos at a file (representation) level instead of at a title (content) level [28]. Thus, in this work, we assume that cache policies are defined at the granularity of files (representations) and consider a cache hit only if the cached file matches the content and representation expected by the user (according to its computing and network resources). To reflect this situation, we adapt the CHR metric by introducing a binary variable z_{u,v_r} that assumes the value 1 only when the cached video matches perfectly with the content and representation expected by the user, and 0 otherwise. Thus, our cache hit ratio is given by:

$$CHR = \frac{\sum_{u \in \mathcal{U}} \sum_{v_r \in \mathcal{I}'} z_{u,v_r} \cdot P_u^{Req}(v_r)}{\sum_{u \in U} \sum_{v_r \in \mathcal{I}} P_u^{Req}(v_r)}. \tag{21}$$

5.5 Evaluating Content, UE Capabilities and Network Conditions

In this section, we evaluate the performance of RAViR and the JCRP variants considering the entire input needed for the model, i.e., video content, user mobile device capabilities, and the quality of the UE communication (including both, UE wireless link and Internet access). We first present the results related to the QoE, considering a fixed ICI (Section 5.5.1). Then, we present the results related to the CHR (Section 5.5.2). Finally, we present an analysis on the impact of the ICI on the QoE (Section 5.5.3).

5.5.1 Performance regarding the QoE

The output of RAViR and the JCRP variants is a list with N recommended videos for each user. To understand how these methods perform in relation to the recommendations and how such recommendations affect the overall QoE perceived by the users, we classify the recommendations into four types:

- edge_rec: recommendation whose video is in the cache and factors **F2**, **F3** and **F4** do not occur;
- cloud_rec: recommendation whose video is not in the cache and factors **F3** and **F4** do not occur;

- under_rec: recommendation whose video resolution is below than what could be reproduced by the UE according to the available resources;
- over_rec: recommendation whose video resolution is higher than what the UE is able to reproduce.

For this evaluation, we consider three scenarios, namely:

- Scenario 1: each user receives a list with three recommended videos ($N = 3$) and the distortion tolerance accepted by the users is 1% ($Tol(u) = 1\%, \forall u \in \mathcal{U}$);
- Scenario 2: each user receives a list with three recommended videos ($N = 3$) and the distortion tolerance accepted by the users is 10% ($Tol(u) = 10\%, \forall u \in \mathcal{U}$);
- Scenario 3: each user receives a list with five recommended videos ($N = 5$) and the distortion tolerance accepted by the users is 1% ($Tol(u) = 1\%, \forall u \in \mathcal{U}$).

Fig. 2 shows the percentage of recommendations that fall into each recommendation type (left Y-axis), as well as the normalized average QoE (right Y-axis) achieved by each method, considering the three scenarios previously described. In this evaluation, $ICI = 60\%$, while the total cache capacity (C) varies in each column of graphics, with the values 50, 300, and 700. Each row exhibits the results from one scenario. As expected, by increasing the total cache capacity from 50 to 700, the percentage of recommendations in the cache (edge_rec) increases, while the percentage of recommendations in the cloud (cloud_rec) decreases in all methods, for all scenarios.

When a large percentage of recommendations made by a certain method falls into the under_rec or over_rec type, the QoE is low. Similarly, as the percentage of recommendations falling into the cloud_rec type increases, the QoE decreases. However, the (negative) impact of under_rec and over_rec recommendations on the QoE is higher than that of cloud_rec recommendations. This is illustrated in Fig. 2(a), where the percentage of cloud_rec recommendations issued by RAViR is the highest among all evaluated methods but, despite this fact, the QoE is the highest for this scenario.

Since scenario 2 (Figs. 2(d), 2(e), and 2(f)) has a higher user distortion tolerance in comparison with scenario 1, the methods have more flexibility to accommodate the different user preferences in the cache. As a consequence, more recommendations fall into the edge_rec type and the QoE increases for all methods. On the other hand, scenario 3 (Figs. 2(g), 2(h) and 2(i)), that has a higher number of recommended videos than scenario 1 but the same user distortion tolerance, becomes more dependent on content provided by the cloud. As a consequence, the QoE is slightly lower for all methods in this scenario.

Overall, we observe that the QoE achieved by RAViR is notably higher than the ones achieved by the JCRP variants in all scenarios and regardless of the total cache capacity. This happens because RAViR takes into account computing and network resources at the moment of its recommendations, avoiding the undesirable recommendation types of under_rec and over_rec. The poor performance of all JCRP variants comes from their unawareness of the computing and network resources and the consequent design focused only on the content cache hit. As a consequence, most of the time, JCRP variants recommend videos with inadequate representations (under_rec or over_rec).

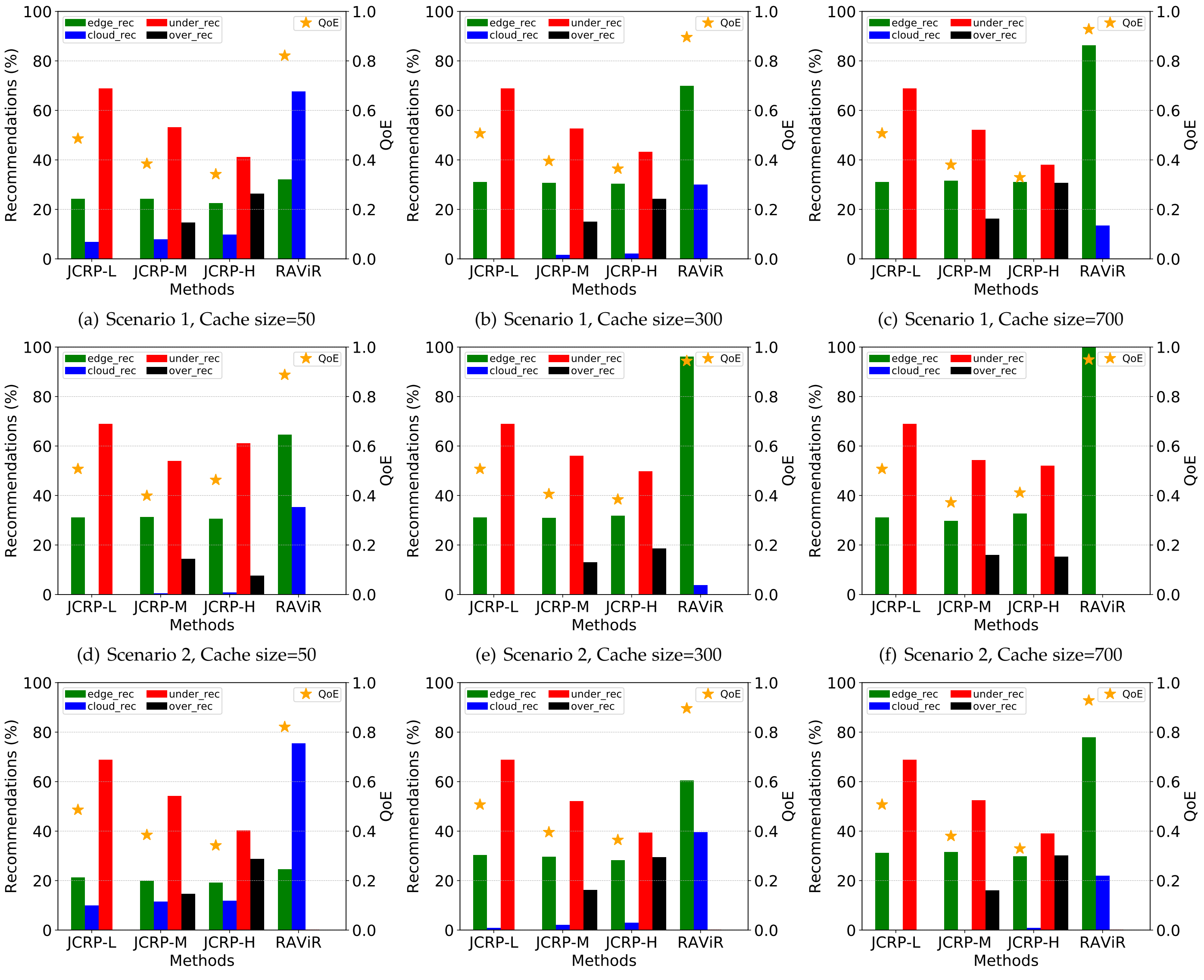
Fig. 3 summarizes the performance of the methods in terms of average user QoE but presenting a larger range of cache capacities (from 50 until 1000). RAViR presents the expected behavior that consists of increasing QoE as a function of the cache capacity. This increase tends to become less relevant as the cache system approaches the optimal condition, i.e., when all videos are cached. Naturally, $Tol(u) = 10\%$ (Fig. 3(b)) presents better results than $Tol(u) = 1\%$ (Fig. 3(a)) because the sharing opportunities are higher. The overall behavior of JCRP-L is similar to RAViR, but exhibiting a notably lower performance since only the lowest representation is cached while many UEs can consume content with higher quality. The behavior of JCRP-M and JCRP-H seems inconsistent since the increase in the cache capacity sometimes implies decreasing QoE. However, this is consistent with the representation unawareness of the method, which implies storing content with representations that can not be consumed by certain UEs due to limitations in its computing or wireless link capacities. This evaluation illustrates numerically the potential benefits of RAViR. For example, RAViR shows an increase from 68% to 85% in QoE when compared with JCRP-L in all cache sizes, and more than 100% when compared with the other methods (JCRP-M and JCRP-H).

5.5.2 Performance regarding the CHR

This evaluation is focused on the cache hit ratio metric (CHR), which was described at the end of Section 5.4. Similar to the previous section, we consider three and five recommendations per user ($N = 3$ and $N = 5$) and the distortion tolerance accepted by the users ($Tol(u)$) assumes 1%. The results obtained with a distortion tolerance of 10% are similar and are omitted due to space constraint.

Fig. 4 shows the performance of the methods in terms of the CHR as a function of the cache capacity, which varies from 50 to 1000. Fig. 4(a) shows the results obtained according to Equation (21), where over_rec and under_rec recommendations are not considered as cache hit. There is some similarity between the CHR and the QoE results shown in Fig. 3(a), although the CHR is more sensitive to the cache capacity. Again, RAViR has a more consistent behavior than the JCRP variants, with the CHR increasing as the cache capacity increases. This consistency is also observed when comparing the results with three and five recommendations per user. As expected, the CHR achieved by RAViR with three recommendations per user is higher than that with five. The poor performance of all JCRP variants was expected due to their unawareness of the computing and network resources. JCRP-L is penalized mostly due to under_rec recommendations, JCRP-M suffers with under and orver_rec recommendations, while JCRP-H is penalized mostly by over_rec recommendations. JCRP-L performs slightly better than the other JCRP variants since it can keep more videos in the cache, as it operates with lower video representations. Overall, the effective CHR achieved by RAViR is at least 14% higher than that of JCRP-L.

Fig. 4(b) shows the same evaluation of Fig. 4(a) but using a more conservative definition for the CHR, where only over_rec recommendations are not considered cache



(a) Scenario 1, Cache size=50

(b) Scenario 1, Cache size=300

(c) Scenario 1, Cache size=700

(d) Scenario 2, Cache size=50

(e) Scenario 2, Cache size=300

(f) Scenario 2, Cache size=700

(g) Scenario 3, Cache size=50

(h) Scenario 3, Cache size=300

(i) Scenario 3, Cache size=700

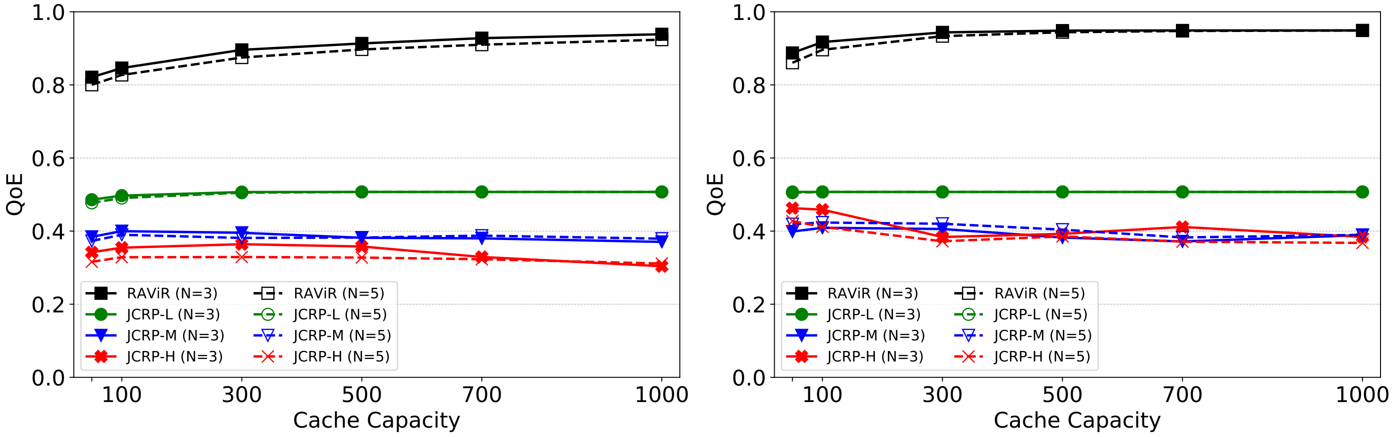
Fig. 2. Percentage of recommendations by type and average user QoE, for each method, in the three considered scenarios. Total cache capacity assume the values 50, 300, and 700.

hit. In this case, when the total cache capacity is small, the JCRP variants perform better than RAViR. This happens because RAViR pursues a perfect match of content and representation for each user, making sharing more difficult among users. For the JCRP variants, however, as long as a cached video matches with the recommended content and does not lead to over_rec recommendation, it can be shared among users. While this more conservative definition of CHR limits the sharing opportunities for RAViR, it also shows, by comparing Figs. 4(b) and 3(a), that high CHR does not imply in high user QoE. Additionally, this conservative definition does not reflect the way DASH (Dynamic Adaptive Streaming over HTTP) clients, commonly used in video streaming, work in practice. In general, DASH clients seek to obtain chunks in the best video representation supported by the UE given the limitations imposed by network throughput.

5.5.3 Impact of the Internet Congestion Index on QoE

To evaluate the impact of the Internet Congestion Index (ICI) on the QoE, we vary this index from 10% to 70%, representing a wide range of network conditions. Naturally, if the recommended video is in the cache, then the UE does not need to access the Internet, i.e., $ICI = 0$. The ICI has a major impact on scenarios where the total cache capacity is small. Thus, this evaluation employs a total cache capacity of 50 ($C = 50$). The experiments are carried out considering three and five recommendations per user ($N = 3$ and $N = 5$), and a distortion tolerance accepted by users of 1% ($Tol(u) = 1, \forall u \in \mathcal{U}$).

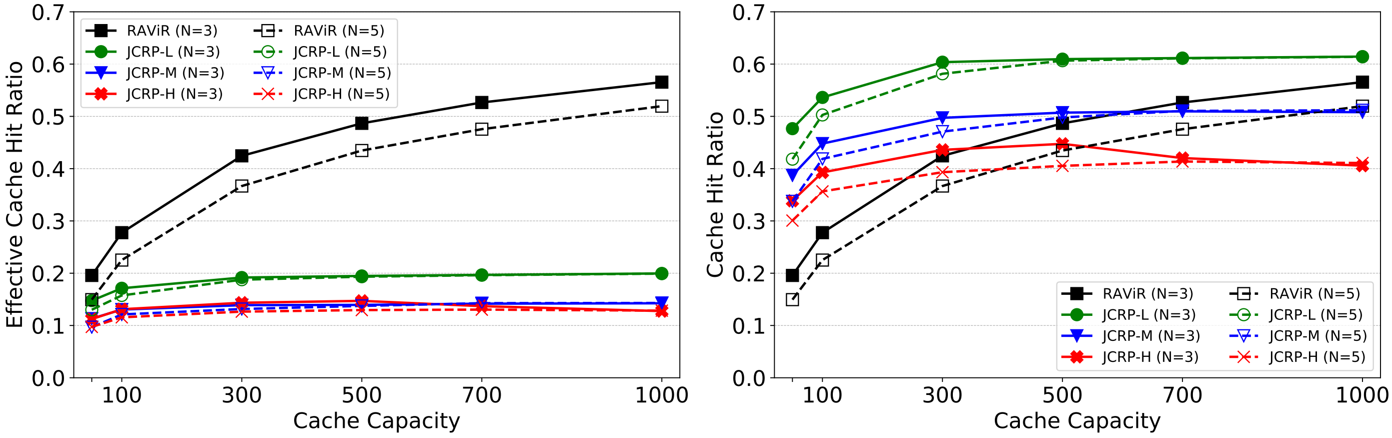
Fig. 5 shows the average QoE as a function of the ICI. On one hand, all JCRP variants are unaware of the network conditions, which makes them less sensitive to the ICI than RAViR. On the other hand, the performance of RAViR is still notably superior to all JCRP variants even in the worst evaluated case. For example, under $ICI = 70\%$, the



(a) $Tol(u) = 1\%$

(b) $Tol(u) = 10\%$

Fig. 3. Impact of the total cache capacity on the average user QoE achieved by RAViR and JCRP variants. In the figures, solid lines represent experiment with 3 recommendations per user ($N = 3$) and dashed lines represents experiment with 5 recommendations per user ($N = 5$).



(a) $Tol(u) = 1\%$

(b) $Tol(u) = 1\%$

Fig. 4. Performance of RAViR and the JCRP variants concerning the CHR for different total cache capacities. In figure (a) over_rec and under_rec recommendations are not considered as cache hit, and in figure (b) over_rec recommendations are not considered as cache hit. Solid lines represent experiment with 3 recommendations per user ($N = 3$) and dashed lines represent experiment with 5 recommendations per user ($N = 5$).

smallest advantage of RAViR ($N = 5$) over JCRP-L ($N = 3$), the JCRP variant with the best performance, is nearly 62%.

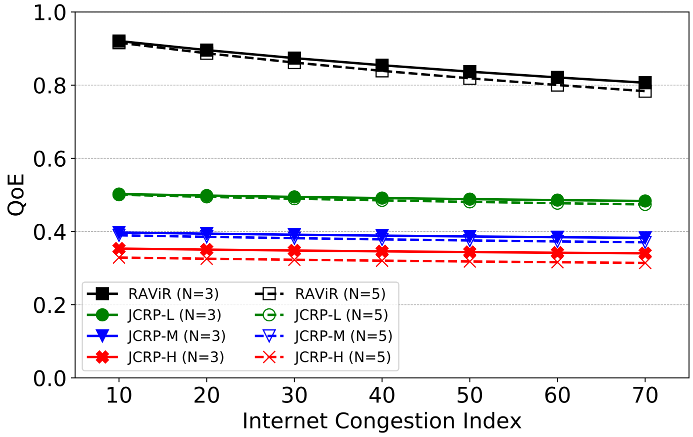


Fig. 5. Impact of the Internet Congestion Index (ICI) on the QoE for RAViR and the JCRP variants. Analysis is conducted considering user distortion tolerance of 1% ($Tol(u) = 1$), total cache capacity of 50 ($C = 50$) and 3 and 5 recommendations per user ($N = 3$ and $N = 5$).

5.6 Highlighting the Network Conditions

Nowadays, user mobile devices are still very heterogeneous and their capabilities to present video content varies significantly. However, the continuous evolution of these devices, in general, followed by large drops in their prices, may change the context in a near future. Under this assumption, the network conditions become the relevant impact factor in the consumption of the video content. In this section, we evaluate the methods assuming that all UEs have computing capabilities to play video content in any resolution, constrained only by the network conditions.

In this context, we need to update some concepts in our model. Initially, we redefine $Best(u)$ (Equation (3)) so that the best video representation for user u is the one that best meets the requirements of the quality of its UE wireless link, i.e.,

$$Best(u) = ResCQI(CQI(u)). \tag{22}$$

Based on the new values of $Best(u)$, we recompute our model (P_u^{Res}, P_u^{Pref}, and P_u^{Req}) and find the new recommendation window W_u, for every $u \in \mathcal{U}$. We then solve the RAViR problem removing the user mobile device constraint. Note that, in this redefinition of the problem, the QoE is now affected by factors **F1**, **F2**, and **F4**. Additionally, the recommendations are now classified into three types: edge_rec, cloud_rec, or under_rec.

Fig. 6 shows the percentage of recommendations that fall into each recommendation type (left Y-axis) and the normalized average QoE (right Y-axis) achieved by each method, considering Scenario 1 described in Section 5.5.1. Scenarios 2 and 3 are omitted for the sake of brevity, but they provide the same conclusions. Figs. 6(a), 6(b), and 6(c) show the result for a total cache capacity of 50, 300, and 700, respectively.

Comparing the results presented in Fig. 2 (from Section 5.5.1) with the results shown in Fig. 6, there is a large decrease in the average QoE achieved by all methods. The reason is the update in the users' expectations that were originally partially represented by their devices capabilities, and are now unconstrained in terms of device resolution (i.e., all UEs can play the highest video resolution). At the same time, the network conditions are the same as the previous evaluation, meaning an increased demand with the same amount of resources (both network and MEC). Thus, the overall decrease in the average QoE is a consistent result. Since video resolution is not constrained in any UE, the JCRP-H variant becomes the one with the best performance, since now it experiences under_rec recommendations only due to the network conditions, and over_rec recommendations do not happen in this context. On the other hand, the network resource awareness of RAViR still provides a remarkable advantage. For example, RAViR offers an average QoE more than 100% higher than all JCRP variants in all evaluated scenarios and cache capacities.

Generally, the users of a Radio Access Network (RAN) experience the same ICI at a specific point in time, but each UE has its own wireless link, i.e., UEs may obtain different CQI values. Thus, CQI is relevant in the choice of the video representation and, as a consequence, in the QoE value. Fig. 7 shows the QoE values as a function of the CQI and video representation. If all representations of all videos were available in the cache system, given a certain CQI value for the UE wireless link, it is expected that a network-aware caching and recommendation method chooses the video representation that provides the highest QoE value (lighters colors in Fig. 7). On the other hand, caching and recommendation methods unaware of network resources may frequently choose video representations under or over the UE wireless link capacity (darker colors in Fig. 7), which impacts negatively in the QoE. Since RAViR is aware of computing and network resources, it operates in the optimal range illustrated in Fig. 7.

6 Conclusions

In this paper, we focus on jointly optimizing cache and recommendations for improving network performance and user satisfaction. We present an optimization model that represents a recommendation system aware of the available resources (cache, resources of the users' devices, and network resources) and coupled with the cache storage policy. We show that taking into consideration the device limitations, video representation, and network conditions in the joint decision on video recommendation and content caching promotes noticeable improvements in user satisfaction. Our proposal, named RAViR, outperforms a state-of-the-art method in all evaluated scenarios, considering both QoE and cache hit ratio metrics.

As future work, we intend to evaluate RAViR in a laboratory testbed, making recommendations and content caching online for synthetic users. In this type of environment, we can evaluate the effective cache hit ratio and also measure other QoE metrics, such as the ITU-T rec. P. 1203 [29]. This sort of experiment also demands some adaptations to the recommendation and cache systems, since the CQI estimator is prone to error and the cached content needs dynamic changes, for example. As a starting point, we are investigating the use of tools such as godash and godashbed [30] to create the testbed. We believe this experimental environment will provide insights for improving RAViR, which we intend to make available as an open-source tool for joint optimization of recommendation and caching in MEC systems of 5G/B5G networks.

Acknowledgments

This work was partially funded by the Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG) in the call No. 03/2015 FAPEG/CNPq/PPP, process No. 201810267001734. This work was also supported by the Ministério da Ciência, Tecnologia e Inovação (MCTI) through the project "Brasil 6G", process No. 01245.010604/2020-14 and by project EMBRACE Inria associated-team.

References

[1] Cisco VNI, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf, 2020.

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

[3] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Jointly Optimizing Content Caching and Recommendations in Small Cell Networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 125–138, 2019.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing: A key technology towards 5G," 2015, ETSI White Paper N. 11.

[5] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, K.-W. Wen, K. Kim, R. Arora, A. Odgers, L. M. Contreras, and S. Scarpina, "MEC in 5G networks," 2018, ETSI White Paper No. 28.

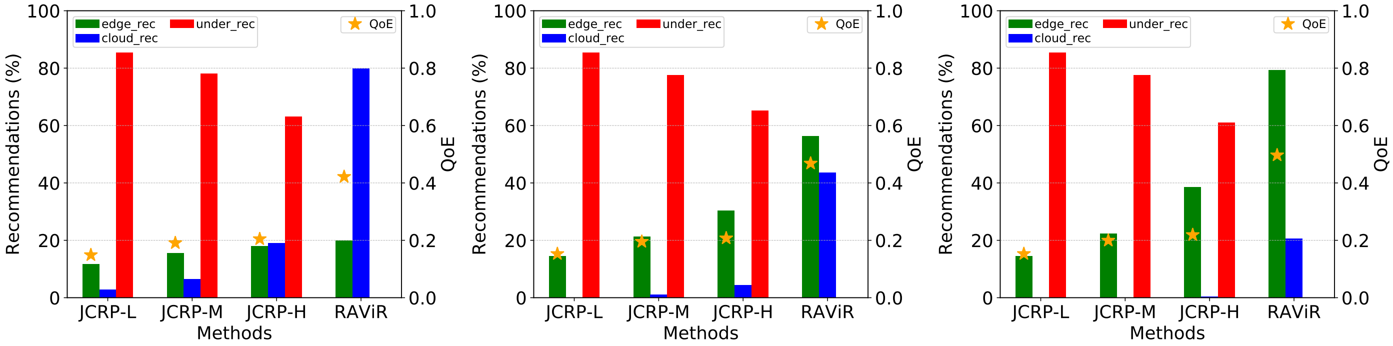
[6] European Telecommunications Standards Institute (ETSI), "Multi-access Edge Computing (MEC); Radio Network Information API," 2019, eTSI GS MEC 012 V2.1.1.

[7] L. M. Contreras, J. Baliosian, P. Martinez-Julia, and J. Serrat, "Computing at the edge, but what edge?" in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2020, pp. 1–9.

[8] F. Malandrino, C.-F. Chiasserini, G. Avino, M. Malinverno, and S. Kirkpatrick, "From Megabits to CPU Ticks: Enriching a Demand Trace in the Age of MEC," pp. 43–50, 2020.

[9] C. A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 13:1–13:19, 2015.

[10] P. Sermpezis, S. Kastanakis, J. I. Pinheiro, F. Assis, M. Nogueira, D. Menasché, and T. Spyropoulos, "Towards qos-aware recommendations," 2020.



(a) Scenario 1, Cache size=50

(b) Scenario 1, Cache size=300

(c) Scenario 1, Cache size=700

Fig. 6. Percentage of recommendations by type and achieved QoE, for each method, in Scenario 1 and total cache capacities of 50, 300 and 700.

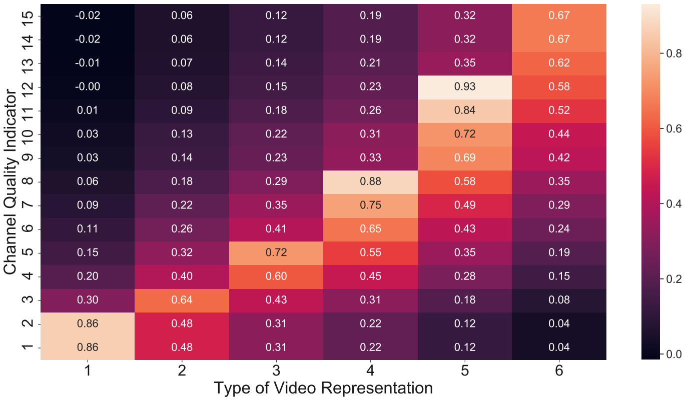


Fig. 7. QoE values as a function of CQI and video representation.

[11] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Dec. 2015. [Online]. Available: https://doi.org/10.1145/2827872

[12] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "A survey of web caching and prefetching," *Int. J. Advance. Soft Comput. Appl*, vol. 3, no. 1, pp. 18–44, 2011.

[13] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1443–1474, 2018.

[14] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-Centric Video Recommendation: An Approach to Improve the Efficiency of YouTube Caches," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 4, 2015.

[15] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, and A. Ashkan, "Cache content-selection policies for streaming video services," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[16] P. Sermpezis, T. Spyropoulos, L. Vigneri, and T. Giannakas, "Femto-Caching with Soft Cache Hits: Improving Performance with Related Content Recommendation," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[17] J. Tadrous, A. Eryilmaz, and H. El Gamal, "Proactive Content Download and User Demand Shaping for Data Networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1917–1930, 2015.

[18] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward A Practical Perceptual Video Quality Metric," https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, 2016, accessed: March-10-2021.

[19] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5g-qoe: Qoe modelling for ultra-hd video streaming in 5g networks," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 621–634, 2018.

[20] K. Saija, S. Nethi, S. Chaudhuri, and R. M. Karthik, "A Machine Learning Approach for SNR Prediction in 5G Systems," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–6.

[21] S. Yang, Y. Tseng, C. Huang, and W. Lin, "Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/qoe models," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1888–1902, 2019.

[22] D. De Vleeschauwer and K. Laevens, "Performance of caching algorithms for iptv on-demand services," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 491–501, 2009.

[23] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: Why it matters and how to model it," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 5, p. 5–12, 2013.

[24] S. Shukla and A. A. Abouzeid, "Proactive retention aware caching," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[25] J. Tadrous and A. Eryilmaz, "On Optimal Proactive Caching for Mobile Networks With Demand Uncertainties," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, p. 2715–2727, 2016.

[26] R. Zhou, S. Khemmarat, and L. Gao, "The Impact of YouTube Recommendation System on Video Views," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010, pp. 404–410.

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

[28] M. Vora, L. Deek, and E. Livengood, "Content Popularity for Open Connect," Available at https://netflixtechblog.com/content-popularity-for-open-connect-b86d56f613b, 2017, accessed: March-10-2021.

[29] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M.-N. Garcia, K. Yamagishi, and S. Broom, "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, p. 466–471.

[30] J. O'Sullivan, D. Raca, and J. J. Quinlan, "Godash 2.0 - The Next Evolution of HAS Evaluation," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2020, pp. 185–187.

Ana Claudia B. L. Monção is an assistant professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where she has been a professor since 2013. She holds a degree in Data Processing Technologist from Pontifical Catholic University of Rio de Janeiro (PUC-Rio) (1992), has MSc in Computer Sciences from Universidade Federal de Goiás (2013) and she is currently working toward the PhD degree in Universidade Federal de Goiás. Her main research interests include wireless networks, caching, recommender systems and data mining.

Sand Luz Correa is an associate professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where she has been a professor and researcher since 2010. She holds a degree in Computer Science from Universidade Federal de Goiás (1994), has MSc (1997) in Computer Sciences from University of Campinas (UNICAMP) and PhD (2011) in Informatics from Pontifical Catholic University of Rio de Janeiro (PUC-Rio). In 2015, she spent her sabbatical at Virginia Tech (in the USA) and, in 2020, at Inria Saclay Research Centre (in France). Professor Sand has participated in some international research projects (including two from joint calls BR-EU). Her research interests include cloud computing, SDN, data plane programmability, and sensor systems and smart city platforms.

Kleber Vieira Cardoso is an associate professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where he has been a professor and researcher since 2009. He holds a degree in Computer Science from Universidade Federal de Goiás (1997), has MSc (2002) and PhD (2009) in Electrical Engineering from COPPE - Universidade Federal do Rio de Janeiro. In 2015, he spent his sabbatical at Virginia Tech (in the USA) and, in 2020, at Inria Saclay Research Centre (in France). Professor Kleber has participated in some international research projects (including two from joint calls BR-EU) and coordinated several national-sponsored research and development projects. His research is focused on the following topics: wireless networks, software-defined networks, virtualization, resource allocation, and performance evaluation.

Aline Carneiro Viana is a permanent Reseach Director (DR) at Inria, where she leads the team TRiBE team. After a 1-year sabbatical leave at the TKN Group of the TU-Berlin, Germany, she got her habilitation degree from UPMC - Sorbonne Universités, France in 2011. Dr. Viana got her PhD in Computer Science from the UPMC - Sorbonne Universités in 2005. Her research addresses the design of solutions for tactful networking, smart cities, mobile and self-organizing networks with the focus on human behavior analysis. She is a recipient of the French Scientific Excellence award since 2015 and for 6 years now and was nominated in 2016 as one of the ”10 women in networking/communications that you should Watch” (1st-year nomination of N2Women community).