Cache-Aided Interactive Multiview Video Streaming in Small Cell Wireless Networks

Eirina Bourtsoulatze*† and Deniz Gündüz*
*Department of Electrical and Electronic Engineering, Imperial College London
†Department of Electronic and Electrical Engineering, University College London
Email:{e.bourtsoulatze, d.gunduz}@imperial.ac.uk

***Abstract*—The emergence of novel interactive multimedia applications with high data rate and low latency requirements has led to a drastic increase in the video data traffic over wireless cellular networks. Endowing the small base stations of a macro-cell with caches that can store some of the contents is a promising technology to cope with the increasing pressure on the backhaul connections, and to reduce the delay for demanding video applications. In this work, delivery of an interactive multiview video over an heterogeneous cellular network is studied. Differently from existing works that focus on the optimization of the delivery delay and ignore the video characteristics, the caching and scheduling policies are jointly optimized, taking into account the quality of the delivered video and the video delivery time constraints. We formulate our joint caching and scheduling problem via submodular set function maximization and propose efficient greedy approaches to find a well performing joint caching and scheduling policy. Numerical evaluations show that our solution significantly outperforms benchmark algorithms based on popularity caching and independent scheduling.**

I. Introduction

Video traffic, which already constitutes a significant portion of the total mobile data traffic, is expected to grow further with the emergence of new multimedia applications, such as virtual reality (VR), augmented reality (AR) and interactive multiview video streaming (IMVS). These novel multimedia technologies offer users the possibility to interact with the application in real time, which, however, imposes more demanding latency and bandwidth requirements that must be met by mobile data operators. To deal with the ever increasing amount of mobile video data, the use of small cell base stations (SBSs) equipped with caches has been proposed [1]. SBS caches can be exploited by placing popular contents during off-peak hours, and serving users locally through short range communication during peak-hours. In that way, the use of costly backhaul links that connect the SBSs to the core network during the peak-hours can be alleviated, and the load on the macro cell base station (MBS) can be reduced [2].

This work has been funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 750254 and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 677854.

In this work, we propose a novel framework for jointly optimizing the caching and scheduling policies for the delivery of IMVS content over a wireless cellular network consisting of a MBS and multiple SBSs equipped with caches. IMVS enables users to freely explore the 3D scene of interest from different viewpoints in real time [3]. To offer such interactivity and enable low-latency view switching at high quality, multiple video streams, each corresponding to a different view, must be delivered to users. Thus, an IMVS application typically requires much higher bandwidth than single view video streaming.

The optimal selection of the delivered set of views, assuming the available bandwidth is known, has been studied in [3]. In our scenario, users' bandwidth resources may vary depending on the density of the SBS placement and the number of users served by these SBSs; and therefore, must be allocated jointly with the caching policy. Also the key objective in the context of caching for real-time video streaming is different from that for video-on-demand (VoD) applications [1], [2], [4], [5]. In the latter, the users request a single file according to some popularity distribution, and the aim is to place the video contents to the caches in a way to minimize the average download delay. This objective is not suitable for real-time video streaming applications, as it ignores the video delivery time constraints and the quality of the delivered content. Differently from the state-of-the-art, our framework takes into account the time constraints of the streamed video and the quality of the video content delivered to the users.

The goal of our joint caching and scheduling policy is to optimally cache and deliver subsets of views in an IMVS application in order to minimize the average expected distortion of the users who can freely navigate through the available set of views during a streaming session. We formulate our problem as the maximization of the reduction in the average expected distortion, in which the objective function is shown to be submodular. In order to efficiently solve the formulated optimization problem, we adopt a greedy algorithm, and show through numerical evaluation that our joint caching and scheduling algorithm significantly outperforms benchmark algo-

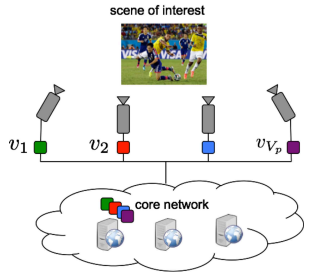

Fig. 1. IMVS system with V_p cameras capturing the scene of interest from multiple viewpoints. Captured content is transmitted to the core network.

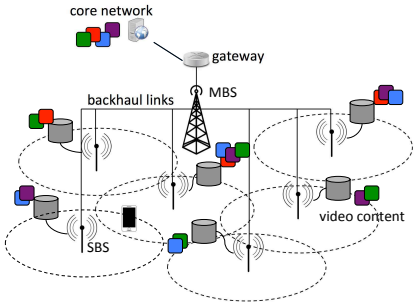

Fig. 2. The multiview video is cached in the small-cell base stations and delivered to wireless users simultaneously when the streaming session begins.

rithms based on popularity caching and independent rate allocation.

II. System Model

We consider an IMVS application illustrated in Fig. 1. An array of equally spaced cameras capture the scene of interest from multiple viewpoints. Let $\mathcal{V}_p = \{v_1, v_2, \ldots, v_{V_p}\}$ be the set of captured views, where $|\mathcal{V}_p| = V_p > 2$. Each camera encodes its view independently, and transmits it to the core network, where the compressed video streams are stored at the content provider's servers, to be further delivered to a set of wireless users (Fig. 2).

Virtual views between two adjacent views $v_i,\ v_{i+1} \in \mathcal{V}_p$ can be synthesized at the decoder, *e.g.*, via the depth image based rendering (DIBR) technique [6]. We denote the set of virtual views as $\mathcal{V}_s$. In order to virtually synthesize a view $v \in \mathcal{V} \triangleq \mathcal{V}_p \cup \mathcal{V}_s$, a left ($v_l$) and a right ($v_r$) reference views ($v_l, v_r \in \mathcal{V}_p$) are required. The distortion at which the virtual view v ($v_l < v < v_r$) is synthesized depends on the quality of the reference views v_l, v_r, and the spatial correlation between v and v_l, v_r. We adopt the distortion model proposed in [7]:

$$d_v(v_l, v_r) = \gamma \mathrm{e}^{\alpha(v_r - v_l)} \left(\mathrm{e}^{\beta \min\{v - v_l, v_r - v\}} - 1 \right), \quad (1)$$

where $d_v(v_l, v_r)$ is the distortion at which view v can be reconstructed from reference views v_l and v_r, and α, β, γ are video related parameters.[1]

[1]This model has been chosen due to its simplicity and accuracy. Our optimization framework is general and can incorporate other distortion models.

We study a streaming scenario in which the IMVS session is initiated some time after the video content is recorded, and focus on the operation of a single macro-cell. Let $\mathcal{U} = \{1, 2, \ldots, U\}$ denote a set of U wireless users served by the macro-cell. The macro-cell consists of $N + 1$ base stations (BSs) in total: an MBS indexed by $n = 0$, and a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of N SBSs located across the macro-cell. The MBS can communicate with all the users within the cell, while the nth SBS is assumed to serve only a subset $\mathcal{U}_n$ of users located within its proximity. We denote the set of BSs that serve user u as $\mathcal{N}_u$. We assume that the MBS and the SBSs are assigned disjoint sets of subchannels, while the neighbouring SBSs operate in orthogonal frequency bands [1]. This permits us to ignore any interference among the BSs. The total transmission capacity of each BS is assumed to be limited, and equal to R_n Mbps, and can be allocated among the users $\mathcal{U}_n$.

SBS n is equipped with a cache of size C_n bytes. Since the video content is available before its dissemination to the users, part of the pre-recorded video content can be placed into the SBS caches during low-traffic hours. During the streaming session, the cached contents can then be directly delivered from the local caches to the users, avoiding the use of backhaul links, and as a result, reducing the delivery delay. To facilitate caching and delivery, the video stream from each camera is partitioned into T segments of b^t bytes, $\forall t \in \mathcal{T} = \{1, 2, \ldots, T\}$.[2] We denote the t-th segment of view v as $B^{v,t}$. The index t also stands for the time slot in which the segment $B^{v,t}$ can be scheduled for delivery.

During the IMVS session a user can select any of the actual camera viewpoints in the set $\mathcal{V}_p$, or synthesize a virtual view from the set $\mathcal{V}_s$ in real time. To enable such interactivity at the best possible quality, the full set of actual camera views $\mathcal{V}_p$ must be delivered to the user at any given time. However, this is not always possible in a bandwidth-limited system due to the strict delay constraints imposed by the IMVS application. Typically only a subset of actual camera views can be delivered to the users. The subset of views delivered to the user determines the distortion at which viewpoints selected by the user and not included in the delivered set of views can be reconstructed. Since the views requested by a user at any given time during an IMVS session are not known a priori, the subset of views to be cached in the SBSs and delivered to each user has to be optimized based on a probabilistic model of the popularity of video segments of each view. For each segment $B^{v,t}$, we define a popularity $p^{v,t} \in [0, 1]$, which represents the probability that the t-th segment of view v will be requested for viewing. The content popularity can be learned by the content provider by analyzing the multimedia content

[2]We assume that the views are symmetrically coded. Hence, the size of a segment does not depend on the view index.

[8], or the history of viewing requests [9].

In this work, we aim to find the optimal joint caching and scheduling policy for the multiview video segments $B^{v,t}$ that minimizes the average expected distortion at the wireless users which participate in the IMVS session. In the next section, we provide the formal problem formulation.

III. Joint Caching and Scheduling problem

Let us define the ground set $\mathcal{E}$ as

$$\mathcal{E} \triangleq \{e^{v,t}_{n,\mathcal{A}_n},\ \forall n \in \mathcal{N} \cup \{0\}, \mathcal{A}_n \subseteq \mathcal{U}_n, v \in \mathcal{V}_p,\ t \in \mathcal{T}\}$$

The element $e^{v,t}_{n,\mathcal{A}_n}$ denotes the placement of the segment $B^{v,t}$ in the nth cache and its scheduling for delivery to a subset $\mathcal{A}_n \subseteq \mathcal{U}_n$ of the users served by BS n. Any joint caching and scheduling policy can then be represented by a subset $\mathcal{S} \in \mathcal{E}$.

The distortion function $D^{v,t}_u(\mathcal{S})$ at user u for reconstructing segment $B^{v,t}$ under a caching and scheduling policy $\mathcal{S}$ is:

$$D^{v,t}_u(\mathcal{S}) = \begin{cases} \tilde{D}^{v,t}_u(\mathcal{S})\left(1 - \mathbb{1}_{\{\mathcal{S}\cap\mathcal{F}^{v,t}_u \neq \emptyset\}}\right), & v \in \mathcal{V}_p \\ \tilde{D}^{v,t}_u(\mathcal{S}), & v \in \mathcal{V}_s \end{cases} \tag{2}$$

where $\tilde{D}^{v,t}_u(\mathcal{S})$ is the minimum distortion at which segment $B^{v,t}$ can be reconstructed, and $\mathcal{F}^{v,t}_u \triangleq \{e^{v,t}_{n,\mathcal{A}_n}, \forall n \in \mathcal{N}_u, \mathcal{A}_n \subseteq \mathcal{U}_n, \text{s.t. } u \in \mathcal{A}_n\},\ \forall\ u \in \mathcal{U}, v \in \mathcal{V}_p,\ t \in \mathcal{T}$. Set $\mathcal{F}^{v,t}_u$ represents all possible ways to deliver segment $B^{v,t}$ to user u. The indicator function $\mathbb{1}_{\{c\}}$ is "1" if the condition c is true, and "0" otherwise. In Eq. (2), we distinguish the following two cases. When view v belongs to the set of actual views $\mathcal{V}_p$, the distortion for reconstructing the segment $B^{v,t}$ at user u is 0, if the segment is delivered to user u by at least one of the BSs in $\mathcal{N}_u$. Otherwise, the distortion is equal to the minimum distortion $\tilde{D}^{v,t}_u(\mathcal{S})$. When view v is a virtual view, the segment $B^{v,t}$ is not delivered to user u, and is synthesized using the corresponding segments of the closest left and right views according to the joint caching and scheduling policy $\mathcal{S}$. The minimum distortion $\tilde{D}^{v,t}_u(\mathcal{S})$ at which user u can reconstruct segment $B^{v,t}$ when it is not delivered is

$$\begin{aligned} \tilde{D}^{v,t}_u(\mathcal{S}) = \sum_{v_l<v}\sum_{v_r>v} d_v(v_l,v_r)\cdot \\ \underbrace{\mathbb{1}_{\{\mathcal{S}\cap\mathcal{F}^{v_l,t}_u\neq\emptyset\}}\prod_{v_l<v_{l'}<v}(1-\mathbb{1}_{\{\mathcal{S}\cap\mathcal{F}^{v_{l'},t}_u\neq\emptyset\}})\cdot}_{\text{A}} \\ \underbrace{\mathbb{1}_{\{\mathcal{S}\cap\mathcal{F}^{v_r,t}_u\neq\emptyset\}}\prod_{v<v_{r'}<v_r}(1-\mathbb{1}_{\{\mathcal{S}\cap\mathcal{F}^{v_{r'},t}_u\neq\emptyset\}})}_{\text{B}}. \end{aligned} \tag{3}$$

Expression A in Eq. (3) is equal to "1" if the segment $B^{v_l,t}$ is delivered to user u by at least one BS, and $v'_l < v_l < v$ for all other segments $B^{v'_l,t}$ delivered to user u. Similarly, expression B in Eq. (3) is equal to "1" if the segment $B^{v_r,t}$ is delivered to user u by at least one BS, and $v'_r > v_r > v$ for all other segments $B^{v'_r,t}$ delivered to user u. The product $A \cdot B$ is non-zero only for a unique (v_l, v_r) pair. To guarantee the reconstruction of any view within set $\mathcal{V}$ at a minimum quality, we assume that views v_1 and v_{V_p} are always delivered to users by the MBS.

Our goal is to devise a joint caching and scheduling policy that minimizes the average expected distortion of the wireless users. Equivalently, we can maximize the average expected distortion reduction. To this aim, we define the distortion reduction at user u for reconstructing segment $B^{v,t}$ as:

$$\Delta D^{v,t}_u(\mathcal{S}) = D_{max} - D^{v,t}_u(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{E}, \tag{4}$$

where D_{max} is the maximum distortion experienced by users when the corresponding segment cannot be reconstructed.

When the element $e^{v,t}_{n,\mathcal{A}_n}$ is included in the joint caching and scheduling policy $\mathcal{S}$, the segment $B^{v,t}$ is placed in the cache of BS n consuming a total space of b^t bytes and a rate of $|\mathcal{A}_n|r$ Mbps is allocated by the BS n to transmit it to the users in $\mathcal{A}_n$. Thus, with each element $e^{v,t}_{n,\mathcal{A}_n}$, we associate a caching cost of b^t bytes and a rate cost of $|\mathcal{A}_n|r$ Mbps. We define the cache cost $c_n(\mathcal{S})$ and rate cost $r^t_n(\mathcal{S})$ functions as:

$$c_n(\mathcal{S}) = \sum_{e^{v,t}_{n',\mathcal{A}_{n'}}\in\mathcal{S}} c_n(e^{v,t}_{n',\mathcal{A}_{n'}}), \tag{5}$$

$$r^t_n(\mathcal{S}) = \sum_{e^{v,t'}_{n',\mathcal{A}_{n'}}\in\mathcal{S}} r^t_n(e^{v,t'}_{n',\mathcal{A}_{n'}}) \tag{6}$$

where $c_n(e^{v,t}_{n',\mathcal{A}_{n'}}) = b^t$ if $n' = n$, and 0 otherwise, and $r_n(e^{v,t}_{n',\mathcal{A}_{n'}}) = |\mathcal{A}_n|r$ if $n' = n, t' = t$, and 0 otherwise. The constant r denotes the video rate. Since we consider symmetric coding of the views, the video rate is the same for all captured views. We also define the cost function $f^{v,t}_n(\mathcal{S})$ as:

$$f^{v,t}_n(\mathcal{S}) = \sum_{e^{v',t'}_{n',\mathcal{A}_{n'}}\in\mathcal{S}} f^{v,t}_n(e^{v',t'}_{n',\mathcal{A}_{n'}}) \tag{7}$$

where $f^{v,t}_n(e^{v',t'}_{n',\mathcal{A}_{n'}}) = 1$ if $n' = n, v' = v, t' = t$, and 0 otherwise. Essentially, function $f^{v,t}_n(\mathcal{S})$ counts the number of times segment $B^{v,t}$ is placed in the cache of BS n.

Equipped with the above notation and definitions, we can now write our joint caching and scheduling optimization problem as follows:

$$\mathcal{S}_{OPT} = \arg\max_{\mathcal{S}\subseteq\mathcal{E}} \frac{1}{U}\frac{1}{T}\sum_{u\in\mathcal{U}}\sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}}\Delta D^{v,t}_u(\mathcal{S})p^{v,t} \tag{8}$$

$$\text{s.t. } c_n(\mathcal{S}) \leq C_n,\ \forall n \in \mathcal{N}, \tag{9}$$

$$r^t_n(\mathcal{S}) \leq R_n,\ \forall n \in \mathcal{N}\cup\{0\},\ t \in \mathcal{T}, \tag{10}$$

$$f_n^{v,t}(\mathcal{S}) \leq 1,\ \forall n \in \mathcal{N} \cup \{0\},\ \forall v \in \mathcal{V}_p\},\ \forall t \in \mathcal{T}, \quad (11)$$

$$e_{0,\mathcal{U}_0}^{v_1,t},\ e_{0,\mathcal{U}_0}^{v_{V_p},t} \in \mathcal{S},\ \forall t \in \mathcal{T}. \quad (12)$$

Constraint (9) is the cache capacity constraint and upper bounds the total amount of data stored in an SBS's cache. Constraint (10) is the transmission capacity constraints, which states that the total rate of content delivered by an SBS in each time slot t must not exceed its transmission capacity. Constraint (11) guarantees that the segment $B^{v,t}$ will be placed in the cache of BS n only once. This constraint is necessary since neither the cache cost function nor the rate cost function can distinguish between two elements $e_{n,\mathcal{A}_n}^{v,t}$ and $e_{n,\mathcal{A}'_n}^{v,t}$ associated with the same segment $B^{v,t}$. In other words, for two elements $e_{n,\mathcal{A}_n}^{v,t}$ and $e_{n,\mathcal{A}'_n}^{v,t} \in \mathcal{S}$ the required cache space calculated by the cache cost function is $2b^t$, and the required rate calculated by the rate cost function is $(|\mathcal{A}_n| + |\mathcal{A}'_n|)r$. In practice, however, the two elements $e_{n,\mathcal{A}_n}^{v,t}$ and $e_{n,\mathcal{A}'_n}^{v,t}$ can be replaced with an equivalent element $e_{n,\mathcal{A}_n \cup \mathcal{A}'_n}^{v,t}$. Hence, the actual cache space needed is b^t, and the actual rate needed is $|\mathcal{A}_n \cup \mathcal{A}'_n|r$. Constraint (11) ensures that only a unique element $e_{n,\mathcal{A}_n}^{v,t}$ for segment $B^{v,t}$ will be included in the solution set. Finally, constraint (12) ensures that all the users can reconstruct any segment at a minimum quality.

IV. Greedy Algorithms

It can be shown that the optimization problem in (8) - (12) has the form of submodular function maximization subject to multiple knapsack constraints [10]. To solve it, we propose two greedy algorithms, namely the *uniform cost greedy algorithm* (UC) [11] and the *weighted cost-benefit greedy algorithm* (WCB) [12]. Both algorithms exploit the submodularity property of the objective function, and select greedily the next element to be included in the solution set.

The UC and WCB greedy algorithms are summarized in Algorithm 1. The UC algorithm takes as input the ground set $\mathcal{E}$, a value query oracle $\Delta D(\mathcal{S})$ that returns the value of the objective function in (8) for some subset $\mathcal{S}$ of the ground set, the cost functions (5), (6) and (7), and the values of the total cache capacity C_n and rate R_n. The WCB algorithm also takes as input the weights λ_1, λ_2, and λ_3. The UC algorithm starts with a solution set $\mathcal{E}_0 = \{e_{0,\mathcal{U}_0}^{v_1,t},\ e_{0,\mathcal{U}_0}^{v_{V_p},t}\}$ (due to constraint (12)), and at the j-th iteration picks the element from the ground set that maximizes the gain with respect to the solution set at step $j-1$ (step 5 of Algorithm 1). If this choice satisfies the problem constraints specified in (9) - (11) and the gain is positive, the element is added to the solution set. Otherwise, the solution set is not updated and the element is removed from the ground set. This procedure is repeated until all elements from the ground set have been either included in the solution set or removed from the ground set. The WCB algorithm works similarly, but instead of selecting the element that maximizes the reduction in distortion, it maximizes a weighted sum of the gain divided by the costs defined in Eqs. (5), (6) and (7), where the weights λ_1, λ_2, and λ_3 satisfy $\lambda_1 + \lambda_2 + \lambda_3 = 1$ (step 5 of the Algorithm 1).

Algorithm 1 UC and WCB greedy algorithms

1: **Input:** $\mathcal{E}$, value query oracle $\Delta D(\mathcal{S})$, C_n, R_n, cost functions c_n, r_n^t, $f_n^{v,t}$, weights λ_1, λ_2, λ_3
2: **Initialization:** $\mathcal{S}_0 \leftarrow \{e_{0,\mathcal{U}_0}^{v_1,t},\ e_{0,\mathcal{U}_0}^{v_{V_p},t}\}$, $k \leftarrow 0$
3: **while** $\mathcal{E} \backslash \mathcal{S}_k \neq \emptyset$ **do**
4: $k \leftarrow k + 1$
5: UC: $e_k \leftarrow \underset{e_{n,\mathcal{A}_n}^{v,t} \in \mathcal{E}_{k-1} \backslash \mathcal{S}_{k-1}}{\arg\max} \Delta_{k-1}(e_{n,\mathcal{A}_n}^{v,t})$

WCB: $e_k \leftarrow \underset{e_{n,\mathcal{A}_n}^{v,t} \in \mathcal{E}_{k-1} \backslash \mathcal{S}_{k-1}}{\arg\max} \lambda_1 \frac{\Delta_{k-1}(e_{n,\mathcal{A}_n}^{v,t})}{c_n(e_{n,\mathcal{A}_n}^{v,t})} + \lambda_2 \frac{\Delta_{k-1}(e_{n,\mathcal{A}_n}^{v,t})}{r_n(e_{n,\mathcal{A}_n}^{v,t})} + \lambda_3 \frac{\Delta_{k-1}(e_{n,\mathcal{A}_n}^{v,t})}{f_n^{v,t}(e_{n,\mathcal{A}_n}^{v,t})}$

$\theta_k \leftarrow \Delta_{k-1}(e_k)$
where $\Delta_{k-1}(e) = \Delta D(\mathcal{S}_{k-1} \cup e) - \Delta D(\mathcal{S}_{k-1})$
6: **if** $c_n(\mathcal{S}_k) \leq C_n$, $r_n^t(\mathcal{S}_k) \leq R_n$, $f_n^{v,t}(\mathcal{S}_k) \leq 1$ and $\theta_k > 0$ **then**
7: $\mathcal{S}_k \leftarrow \mathcal{S}_{k-1} \cup e_{n,\mathcal{A}_n}^{v,t}$
8: **else**
9: $\mathcal{E} \leftarrow \mathcal{E} \backslash e_{n,\mathcal{A}_n}^{v,t}$
10: **end if**
11: **end while**
12: **Output:** $\mathcal{S} \leftarrow \mathcal{S}_k$

V. Performance evaluation

We consider a circular cell consisting of an MBS with transmission range 400m located at the centre, 20 SBSs with coverage radius 100m placed uniformly at random within the cell and 200 wireless users uniformly distributed across the macro cell. The transmission capacity of the SBSs is set to 100Mbps.

The IMVS system consists of $V_p = 8$ cameras. Each view is encoded at $r = 2$ Mbps and divided into $T = 20$ segments of equal size. We assume that $L = 3$ virtual views can be synthesized between every two adjacent physical views. The users select the first segment among the captured views uniformly at random. Then, during the streaming session, each user can switch from view v_i to a neighbouring actual or virtual view v_j with probability $p(v_j|v_i) \propto \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(v_j - v_i)^2}{2\sigma^2}}$ for $|v_j - v_i| \leq W$, and 0 otherwise. We set $W = 8$ and $\sigma^2 = 5/(L+1)$. From this model, we calculate the popularity distribution $p^{v,t}$ of the video segments.

We compare the proposed greedy joint caching and scheduling algorithms with a maximum popularity based caching algorithm. The latter fills each SBS's cache with the most popular video segments. It then performs

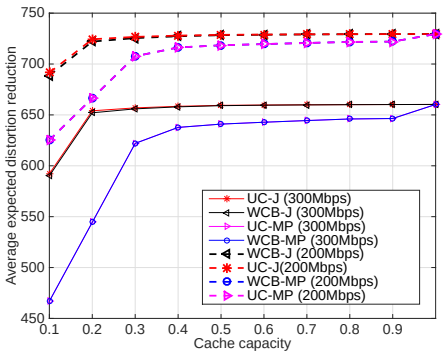

Fig. 3. Average expected distortion reduction vs the cache capacity.

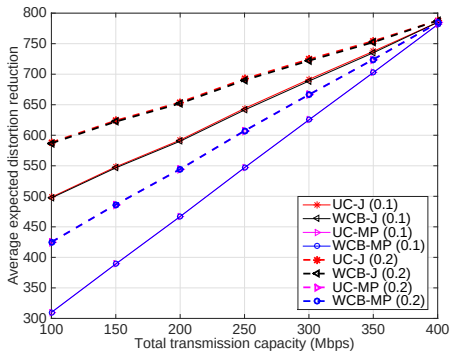

Fig. 4. Average expected distortion reduction vs the total transmission rate of the MBS.

greedy scheduling independently of the cache placement phase.

Fig. 3 shows the average expected distortion versus the cache capacity of the SBSs expressed as a percent of the total size of the multiview video. UC-J and WCB-J denote the uniform cost and weighted cost-benefit greedy algorithms, respectively, for joint caching and scheduling. UC-MP and WCB-MP denote the maximum popularity caching algorithm with uniform cost and weighted cost-benefit greedy scheduling, respectively. We present results for a total transmission capacity of the MBS equal to 200Mbps and 300Mbps. For the WCB algorithm we have used $\lambda_1 = 0.2$, $\lambda_2 = 0.5$ and $\lambda_3 = 0.3$. The results indicate that the joint caching and scheduling algorithms outperform the maximum popularity counterparts for all values of the cache capacity. For low values of the cache capacity, the difference in the performance is significant as the maximum popularity algorithm caches the same content in all SBSs; thus the content diversity across the network is limited. Along with the most popular content cached only in few SBSs, the joint caching and scheduling algorithm also caches the less popular contents, which, when delivered to the users, improves the reconstruction quality of the views. It is worth noting that this range of capacity values is of great practical interest, as SBSs are typically assumed to cache only 5-10% of the total video catalogue [13], [14]. The performance of all the algorithms becomes limited by the insufficient transmission capacity of the network. Thus, even though all the SBSs can cache almost all of the contents, they cannot be delivered to the users. Finally, we can see that the uniform cost and weighted cost-benefit greedy algorithms perform identically. This is because all the video segments have the same size in this example. We expect that in the case of multiple multiview videos encoded at different rates, or even asymmetrically encoded views, the performance of the two algorithms would be different. We leave this investigation for our future work.

In Fig. 4 we illustrate the average expected distortion versus the total transmission capacity of the MBS for cache capacity equal to 10% and 20% of the total size of the video. As the total transmission capacity of the MBS increases, the average expected quality of the multview video delivered to the users improves. We can see that the joint caching and scheduling algorithm outperforms the maximum popularity algorithm for all values of the MBS transmission capacity. Although the cache capacity of the SBSs is limited, our algorithm performs much better compared to the maximum popularity algorithm due to the more efficient use of the available cache and transmission capacities. As previously, the content diversity is higher when the caching and scheduling policies are optimized jointly. It is worth noting that to achieve the same average expected distortion reduction, the maximum popularity caching algorithm requires a much higher transmission rate to be allocated by the MBS compared to the case of joint cache and scheduling optimization.

VI. Conclusions and Future Work

In this paper, we have presented a framework for jointly optimizing the caching and scheduling policy for interactive multiview video delivery over a wireless cellular network. Unlike existing works for wireless edge caching, our scheme takes into account the quality of the video delivered to the users and the rate requirements for real-time video delivery. Numerical evaluation of our scheme shows that the joint policy performs significantly better than the independent caching and scheduling policies for the case of multiview video. In our future work, we will investigate ways to simplify the expression for calculating the distortion of the delivered video, with the aim of obtaining a convex problem which can be solved for optimality.

References

[1] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," in *IEEE INFOCOM'12*, Mar. 2012.

[2] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Trans. on Mobile Comp.*, vol. 16, no. 3, pp. 675–687, Mar. 2017.

[3] A. D. Abreu, L. Toni, N. Thomos, T. Maugey, F. Pereira, and P. Frossard, "Optimal layered representation for adaptive interactive multiview video streaming," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 255 – 264, 2015.

[4] E. Ozfatura and D. Gunduz, "Mobility and popularity-aware coded small-cell caching," *IEEE Communications Letters*, vol. 22, no. 2, pp. 288–291, Feb. 2018.

[5] A. Khreishah, J. Chakareski, and A. Gharaibeh, "Joint Caching, Routing, and Channel Assignment for Collaborative Small-Cell Cellular Networks," *IEEE JSAC*, vol. 34, no. 8, pp. 2275–2284, Aug. 2016.

[6] M. Schmeing and X. Jiang, *Depth Image Based Rendering*. Springer Berlin Heidelberg, 2011, pp. 279–310.

[7] D. Ren, S.-H. G. Chan, G. Cheung, V. Zhao, and P. Frossard, "Collaborative P2P streaming of interactive live free viewpoint video," *arXiv:1211.4767v1 [cs.MM]*, 2012.

[8] C. Tzelepis, Z. Ma, V. Mezaris, B. Ionescu, I. Kompatsiaris, G. Boato, N. Sebe, and S. Yan, "Event-based media processing and analysis: A survey of the literature," *Image and Vision Computing*, vol. 53, pp. 3 – 19, 2016.

[9] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *Proc. of IEEE Int'l Conf. on Communications (ICC)*, 2014.

[10] E.Bourtsoulatze and D. Gündüz, "Cache-aided interactive multiview video streaming in small cell wireless networks," *arXiv:1804.01035 [cs.NI]*, 2018.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, New York, NY, 2007, pp. 420–429.

[12] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "Delay-power-rate-distortion optimization of video representations for dynamic adaptive streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[13] L. Maggi, L. Gkatzikis, G. Paschos, and J. Leguay, "Adapting caching to audience retention rate: Which video chunk to store?" *arXiv:1512.03274v1 [cs.NI]*, 2012.

[14] G. Parisis, V. Sourlas, K. V. Katsaros, W. K. Chai, G. Pavlou, and I. Wakeman, "Efficient content delivery through fountain coding in opportunistic information-centric networks," *Comput. Commun.*, vol. 100, no. C, pp. 118–128, Mar. 2017.