Femto-Caching with Soft Cache Hits: Improving Performance through Recommendation and Delivery of Related Content

Pavlos Sermpezis[1], Thrasyvoulos Spyropoulos[2], Luigi Vigneri[2], and Theodoros Giannakas[2]
[1] ICS-FORTH, Greece, sermpezis@ics.forth.gr
[2] EURECOM, France, first.last@eurecom.fr

***Abstract*—Pushing popular content to cheap "helper" nodes (e.g., small cells) during off-peak hours has recently been proposed to cope with the increase in mobile data traffic. User requests can be served locally from these helper nodes, if the requested content is available in at least one of the nearby helpers. Nevertheless, the collective storage of a few nearby helper nodes does not usually suffice to achieve a high enough hit rate in practice. We propose to depart from the assumption of hard cache hits, common in existing works, and consider "soft" cache hits, where if the original content is not available, some related contents that are locally cached can be recommended instead. Given that Internet content consumption is entertainment-oriented, we argue that there exist scenarios where a user might accept an alternative content (e.g., better download rate for alternative content, low rate plans, etc.), thus avoiding to access expensive/congested links. We formulate the problem of optimal edge caching with soft cache hits in a relatively generic setup, propose efficient algorithms, and analyze the expected gains. We then show using synthetic and real datasets of related video contents that promising caching gains could be achieved in practice.**

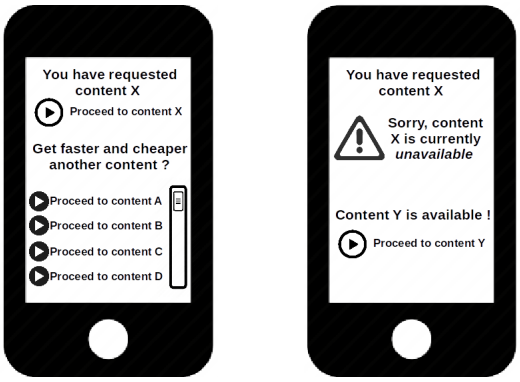
I. Introduction

Mobile edge caching has been identified as one of the five most disruptive enablers for 5G networks [1], both to reduce content access latency and to alleviate backhaul congestion. However, the number of required storage points in future cellular networks will be orders of magnitude more than in traditional CDNs [2] (e.g., 100s or 1000s of small cells corresponding to an area covered by a single CDN server today). As a result, the storage space per local edge cache must be significantly smaller to keep costs reasonable. Even if we considered a small subset of the entire Internet catalogue, e.g., a typical torrent catalog (1.5PB) or the Netflix catalogue (3PBs), with relatively skewed popularity distribution, and more than 1TB of local storage, cache hit ratios would still be low [3], [4].

Additional caching gains have been sought by researchers, increasing the "effective" cache size visible to each user. This could be achieved by: (a) *Coverage overlaps*, where each user is in range of multiple cells, thus having access to the aggregate storage capacity of these cells, as in the femto-caching framework [5], [6]. (b) *Coded caching*, where collocated users overhearing the same broadcast channel may benefit from cached content in other users' caches [7]. (c) *Delayed content access*, where a user might wait up to a TTL for its request, during which time more than one cache (fixed [8] or mobile [9], [10], [11]) can be encountered. Each of these ideas could theoretically increase the cache hit ratio (sometimes significantly), but the actual practical gains might not suffice by themselves (e.g., due to high enough cell density required for (a), sub-packetization complexity in (b), and imposed delays in (c)).

Existing edge caching approaches have a common goal: to deliver *every possible* content a user requests (if not from a local cache, then from a remote content server). While reasonable, this leads to many expensive cache misses that may potentially stifle the idea of edge caching. We argue that, in an Internet which is becoming increasingly entertainment-oriented *moving away from satisfying a given user request towards satisfying the user* could prove beneficial for caching systems. When a user requests a content not available in the local cache(s), a recommendation system could propose a set of *related contents* that *are* locally available. If the user accepts one of these contents, an expensive remote access could be avoided. We will use the term *soft cache hit* to describe such scenarios.

Although many users in today's cellular ecosystem might be reluctant to accept alternative contents, we believe there are a number of scenarios where soft cache hits could benefit both the user and the operator. As one example, a *cache-aware* recommendation system could be a plugin to an existing application (e.g., the YouTube app), as shown in Fig. 1(a). The operator can give incentives to users to accept the alternative contents when there is congestion (e.g., *zero-rating* services [12], [13]) or letting the user know that accessing content X from the core infrastructure would be slow and choppy, while contents $A, B, C, ...$ might have much better performance. The user can still reject the recommendation and demand the original content. In a second example (see Fig. 1(b)) the operator might "enforce" an alternative (but related) content, e.g.: (i) making very low rate plans (currently offering little or no data) more interesting by allowing regular data access, except under congestion, at which time only locally cached content can be served; (ii) in developing areas [14] or when access to only a few Internet services is provided, e.g., the Facebook's Internet.org (Free Basics)



(a) (b)

Fig. 1: Mobile app example for *Soft Cache Hits*: (a) related content *recommendation* (that the user might not accept) , and (b) related content *delivery*.

project [15], [16].

We believe such a system is quite timely, given the increased convergence of content providers with sophisticated recommendation engines (e.g., NetFlix and YouTube) and Mobile Network Operators (MNO), in the context of RAN Sharing [17], [18]. Furthermore, the idea of soft cache hits is complementary and can be applied *on top* of existing proposals for edge caching, like the ones described earlier. In a recent preliminary work [19], we have considered the idea of soft cache hits in a DTN context with mobile relays. Our goal in this paper is to develop the idea of soft cache hits in detail, applying it to standard mobile edge caching systems with cache cooperation (e.g., [5]). To our best knowledge, this is the first work to jointly consider related content recommendation/delivery gains and cache cooperation (e.g., femto-caching) gains. In this context, our main contributions are:

- *Soft Cache Hits (SCH) model:* We propose a generic model for mobile edge caching and alternative soft cache hits that can capture a number of interesting scenarios (both Fig. 1(a) and Fig. 1(b) - in Sections II and V, respectively).
- *Single cache with SCH:* We formulate the problem of edge caching with SCH, in the context of a single cache. We show that the problem is NP-hard, and propose efficient approximation algorithms with provable performance (Section III).
- *Femto-caching with SCH:* We extend our framework to a femto-caching scenario with SCH, where each user might have access to more than one BS and local caches, as in [5] (Section IV).
- *Validation:* We show using both synthetic data and a real dataset of YouTube related videos that additional caching gains, e.g., *on top of what femto-caching provides*, could be achieved in practice (Section VI).

Finally, we discuss related work and future research directions in Section VII, and conclude our paper in Section VIII.

II. Problem Setup

A. Network and Caching Model

Network Model: Our network consists of a set of users $\mathcal{N}$ ($\|\mathcal{N}\| = N$) and a set of small cells (SC) $\mathcal{M}$ ($\|\mathcal{M}\| = M$). Users are mobile and the SCs with which they associate might change over time. Since the caching decisions are taken in advance (e.g., the night before, as in [5], [6], or once per few hours or several minutes) it is hard to know the exact SC(s) each user will be associated at the time she requests a content. To capture user mobility, we propose a more generic model than the fixed bipartite graph of [5]:

$$q_{ij} \doteq Prob\{\text{user i in range of SC j}\}, \tag{1}$$

or, equivalently, the percentage of time a user i spends in the coverage of SC j). Hence, deterministic q_{ij} ($\in \{0,1\}$) captures the static setup of [5], while uniform q_{ij} ($q_{ij} = q, \forall i,j$) represents the other extreme (no advance knowledge).

Content Model: We assume each user requests a content from a catalogue $\mathcal{K}$ with $\|\mathcal{K}\| = K$ contents. A user $i \in \mathcal{N}$ requests content $k \in \mathcal{K}$ with probability p_k^i.[1] We will initially assume that all contents have the same size, and relax the assumption later.

Cache Model (Baseline): We assume that each SC is equipped with storage capacity of C contents (all our proofs hold also for different cache sizes). We use the integer variable $x_{kj} \in \{0,1\}$ to denote if content k is stored in SC j. In the traditional caching model, which we will consider as our baseline, if a user i requests a content k which is stored in some nearby SC, then the content can be accessed directly from the local cache and a *cache hit* occurs. This type of access is considered "cheap", while a *cache miss* leads to an "expensive" access (e.g., over the SC backhaul and core network).

For ease of reference, the notation is summarized in Table I.

B. Soft Cache Hits

Up to this point the above model describes a baseline setup similar to the popular femto-caching framework [5] (where we consider 0-1 cache hits, for simplicity, rather than access delay). The main departure in this paper is the following.

Alternative Content Recommendation: When the content a user asks for is not found in a local cache, a list of related contents out of the ones already cached is recommended to the user (see, e.g., Fig. 1(a)). If a user selects one of them, a *(soft) cache hit* occurs, otherwise there is a *cache miss* and the network must fetch and deliver the original content.[2] Whether a user accepts an alternative content or not depends both on

[1] This also generalizes the standard femto-caching model [5] which assumes same popularity per user. We can easily derive such a popularity *per content* p_k from p_k^i.

[2] Throughout our proofs, we assume, for simplicity, that the user can pick *any* of the available cached contents; however, our analysis holds also when only a small subset of locally cached contents is recommended (e.g., the ones the recommender thinks are the most related for that user and for that original request).

TABLE I: Important Notation

$\mathcal{N}$	set of users ($\|\mathcal{N}\| = N$))
$\mathcal{M}$	set of small cells (SC) ($\|\mathcal{M}\| = M$))
C	storage capacity of a SC
q_{ij}	probability user i in range of SC j (or, q_i for the single-cache case)
$\mathcal{K}$	set of contents ($\|\mathcal{K}\| = K$))
p_k^i	probability user i to request content k
x_{kj}	content k is stored in SC j ($x_{kj} = 1$) or not ($x_{kj} = 0$) (or, x_k for the single-cache case)
u_{kn}^i	utility of content n for a user i requesting content k
$F_{kn}(x)$	distribution of utilities u_{kn}^i, $F_{kn}(x) = P\{u_{kn}^i \leq x\}$
u_{kn}	average utility for content pair $\{k, n\}$ (over all users)
s_k	size of content k

the content (how related it is) and on the user. This is captured in the following:

Definition 1. *A user i that requests a content k that is not available, accepts a recommended content n with probability u_{kn}^i, where $0 \leq u_{kn}^i \leq 1$, and $u_{kk}^i = 1, \forall i, k$.*

The utilities define a content relation matrix $\mathbf{U^i} = \{u_{kn}^i\}$ for each user. Note that the above model captures the scenario of Fig. 1(a). We will use this model throughout Sections III and IV to develop most of our theory. However, in Section V, we will modify our model to also analyze the scenario of Fig. 1(b), which we will refer to as *Alternative Content Delivery*.

Per user utilities u_{kn}^i could be estimated from past statistics, and/or user profiles as usually done by standard recommendation algorithms [20]. In some cases, the system might have a coarser view of these utilities (e.g., item-item recommendation [21]). We develop our theory and results the most generic case of Definition 1, but we occasionally refer to the following two subcases:

Sub-case 1: The system does not know the exact utility u_{kn}^i for each node i, but only how they are distributed among all nodes, i.e., the distributions $F_{kn}(x) \equiv P\{u_{kn}^i \leq x\}$.

Sub-case 2: The system knows only the *average utility* u_{kn} per content pair $\{k, n\}$.

III. Single Cache with Soft Cache Hits

In order to better understand the impact of the related content matrices $\mathbf{U^i}$ on caching performance, we first consider a scenario where a user i is served by a single base station, i.e., $\sum_j q_{ij} = 1, \forall i$ (i.e., each user is associated to exactly one BS, but we might still not know in advance which). Such a scenario is in fact relevant in today's networks, where the cellular network first chooses a single BS to associate a user to (e.g., based on signal strength), and then the user makes its request [22]. In that case, we can optimize each cache independently. We can also drop the second index for both the storage variables x_{kj} and connectivity variables q_{ij}, to simplify notation.

A. Soft Cache Hit Ratio

A request (from a user to the SC) for a content $k \in \mathcal{K}$ would result in a (standard) cache hit only if the node stores content k in the cache, i.e., if $x_k = 1$. Hence, the (baseline) *cache hit ratio* for this request is simply

$$CHR(k) = x_k$$

If we further allow for soft cache hits, the user might be also satisfied by receiving a different content $n \in \mathcal{K}$. The probability of this event is, by Definition 1, equal to u_{kn}^i. The following Lemma derives the total cache hit rate in that case.

Lemma 1 (Soft Cache Hit Ratio (SCHR))**.** *Let $SCHR$ denote the expected cache hit ratio for a single cache (including regular and soft cache hits), among all users. Then,*

$$SCHR = \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1 - \prod_{n=1}^{K} \left(1 - u_{kn}^i \cdot x_n\right)\right). \quad (2)$$

Proof. The probability of satisfying a request for content k by user i with related content n is $P\{n|k,i\} = u_{kn}^i \cdot x_n$, since u_{kn}^i gives the probability of acceptance (by definition), and x_n denotes if content n is stored in the cache (if the content is not stored, then $P\{n|k,i\} = 0$). Hence, it follows easily that the probability of a *cache miss*, when content k is requested by user i, is given by $\prod_{n=1}^{K}(1 - u_{kn}^i \cdot x_n)$. The complementary probability, defined as the *soft cache hit ratio* (SCHR), is then

$$SCHR(i, k, \mathbf{U}) = 1 - \prod_{n=1}^{K}(1 - u_{kn}^i \cdot x_n). \quad (3)$$

Summing up over all users that might be associated with that BS (with probability q_i) and all contents that might be requested (p_k^i) gives us Eq.(2) □

Lemma 1 can be easily modified for the the sub-cases 1 and 2 of Def. 1 presented in Section II-B. We state the needed changes in Corollary 1.

Corollary 1. *Lemma 1 holds for the the sub-cases 1 and 2 of Def. 1, by substituting in the expression of Eq.* (2) *the term u_{kn}^i with*

$$u_{kn}^i \to E[u_{kn}^i] \equiv \int (1 - F_{kn}(x))\, dx \qquad \textit{(for sub-case 1)} \quad (4)$$

$$u_{kn}^i \to u_{kn} \qquad \textit{(for sub-case 2)} \quad (5)$$

Proof. The proof is given in Appendix A. □

B. Optimal SCH for Same Content Sizes

The (soft) cache hit ratio depends on the contents that are stored in a SC. The network operator can choose the storage variables x_k to maximize SCHR by solving the following optimization problem.

Optimization Problem 1. *The optimal cache placement problem for a single cache with soft cache hits and content relations described by the matrix $\mathbf{U} = \{u_{kn}^i\}$, is*

$$\underset{X=\{x_1,\ldots,x_K\}}{\textit{maximize}} \; f(X) = \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1 - \prod_{n=1}^{K} \left(1 - u_{kn}^i \cdot x_n\right)\right), \quad (6)$$

$$\sum_{k=1}^{K} x_k \leq C. \quad (7)$$

Algorithm 1 Greedy Algorithm for Optimization Problem 1. *computation complexity:* $O(C \cdot K)$

Input: *utility* $\{u_{kn}^i\}$, *content demand* $\{p_k^i\}$, *mobility* $\{q_i\}$, $\forall k, n \in \mathcal{K},\ i \in \mathcal{N}$

1: $S_0 \leftarrow \emptyset;\ t \leftarrow 0$
2: **while** $t < C$ **do**
3: $\quad t \leftarrow t + 1$
4: $\quad n \leftarrow \underset{\ell \in K \setminus S_{t-1}}{argmax} f(S_{t-1} \cup \{\ell\})$
5: $\quad S_t \leftarrow S_{t-1} \cup \{n\},$
6: **end while**
7: $S^* \leftarrow S_t$
8: **return** S^*

In the following, we prove that the above optimization problem is NP-hard (Lemma 2), and study the properties of the objective function Eq. (6) (Lemma 3) that allow us to design an efficient approximate algorithm (Algorithm 1) with provable performance guarantees (Theorem 1).

Lemma 2. *The Optimization Problem 1 is NP-hard.*

Lemma 3. *The objective function of Eq.(6) is submodular and monotone.*

The proofs for the previous two Lemmas can be found in Appendices B and C, respectively.

We propose Algorithm 1 as a greedy algorithm for Optimization Problem 1: to select the contents to be stored in the cache, we start from an empty cache (line 1), and start filling it (one by one) with the content that increases the most the value of the objective function (line 4), till the cache is full. The computation complexity of the algorithm is $O(C \cdot K)$, since the loop (lines 2-6) denotes C repetitions, and in each repetition the objective function is evaluated y times, where $K \geq y \geq K - C + 1$.

The following theorem gives the performance bound for Algorithm 1.

Theorem 1. *Let OPT be the optimal solution of the Optimization Problem 1, and S^* the output of Algorithm 1. Then, it holds that*

$$f(S^*) \geq \left(1 - \frac{1}{e}\right) \cdot OPT \tag{8}$$

Proof. Lemma 3 shows that the Optimization Problem 1 belongs to the generic category of maximization of submodular and monotone functions (Eq. (6)) with a cardinality constraint (Eq. (7)). For such problems, it is known that the greedy algorithm achieves (in the worst case) a $\left(1 - \frac{1}{e}\right)$-approximation solution [23], [24]. □

While the above is a strict worst case bound, it is known that greedy algorithms perform quite close to the optimal in most scenarios. In Section VI we show that this simple greedy algorithm can already provide interesting performance gains.

C. Optimal SCH for Different Content Sizes

Till now we have assumed that all contents have equal size. In practice, each content has a different size s_k and the capacity C of each cache must be expressed in Bytes. Additionally, if a user requests a video of duration X and she should be recommended an alternative one of similar duration Y (note that similar duration does not always mean similar size). While the latter could still be taken care of by the recommendation system (our study of a real dataset in Section VI suggests that contents of different sizes might still be tagged as related), we need to revisit the optimal allocation problem: the capacity constraint of Eq.(7) is no longer valid, and Algorithm 1 can perform arbitrarily bad.

Optimization Problem 2. *The optimal cache placement problem for a single cache with soft cache hits and variable content sizes, and content relations described by the matrix* $\mathbf{U} = \{u_{kn}^i\}$ *is*

$$\underset{X=\{x_1,..,x_K\}}{\textit{maximize}} \ f(X) = \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1 - \prod_{j=1}^{K} \left(1 - u_{kn}^i \cdot x_n\right)\right), \tag{9}$$

$$\sum_{k=1}^{K} s_k x_k \leq C. \tag{10}$$

Remark: Note that the objective is still in terms of cache hit ratio, and does not depend on content size. This could be relevant when the operator is doing edge caching to reduce *access latency* to contents (latency is becoming a core requirement in 5G), in which case a cache miss will lead to long access latency (to fetch the content from deep inside the network), for both small and large contents.

The problem is a set cover problem variant with a knapsack type constraint. We propose approximation Algorithm 2 for this problem, which is a "fast greedy" algorithm (based on a modified version of the greedy Algorithm 1) and has complexity $O\left(K^2\right)$.

Theorem 2.
(1) The Optimization Problem 2 is NP-hard.
(2) Let OPT be the optimal solution of the Optimization Problem 2, and S^* the output of Algorithm 2. Then, it holds that

$$f(S^*) \geq \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot OPT \tag{11}$$

Proof. The proof can be found in Appendix D. □

In fact, a polynomial algorithm with better performance $\left(1 - \frac{1}{e}\right)$-approximation could be described, based on [25]. However, the improved performance guarantees come with a significant increase in the required computations, $O\left(K^5\right)$, which might not be feasible in a practical scenario when the catalog size K is large. We therefore just state its existence, and do not consider the algorithm further in this paper (the algorithm can be found in Appendix H).

Algorithm 2 $\frac{1}{2} \cdot \left(1 - \frac{1}{e}\right)$-approximation Algorithm for Optimization Problem 2.
computation complexity: $O\left(K^2\right)$

Input: *utility* $\{u_{kn}^i\}$, *content demand* $\{p_k^i\}$, *content size* $\{s_k\}$, *mobility* $\{q_i\}$, $\forall k, n \in \mathcal{K},\ i \in \mathcal{N}$

$S^{(1)} \leftarrow$ MODIFIEDGREEDY($\emptyset$,[s_1, s_2,...,s_k])
$S^{(2)} \leftarrow$ MODIFIEDGREEDY($\emptyset$,[1, 1,...,1])
if $f(S^{(1)}) > f(S^{(2)})$ **then**
    $S^* \leftarrow S^{(1)}$
else
    $S^* \leftarrow S^{(2)}$
end if
return S^*

function MODIFIEDGREEDY(S_0,[w_1, w_2,...,w_k])
    $\mathcal{K}^{(1)} \leftarrow \mathcal{K};\ c \leftarrow 0;\ t \leftarrow 0$
    while $\mathcal{K}^{(1)} \neq \emptyset$ **do**
        $t \leftarrow t + 1$
        $n \leftarrow \underset{\ell \in K \setminus S_{t-1}}{argmax} \frac{f(S_{t-1} \cup \{\ell\})}{w_\ell}$
        if $c + w_n \leq C$ **then**
            $S_t \leftarrow S_{t-1} \cup \{n\}$
            $c \leftarrow c + w_n$
        else
            $S_t \leftarrow S_{t-1}$
        end if
        $\mathcal{K}^{(1)} \leftarrow \mathcal{K}^{(1)} \setminus \{n\}$
    end while
    return $\leftarrow S_t$
end function

IV. Femtocaching with Related Content Recommendation

Building on the results and analytical methodology of the previous section for the optimization of a single cache with soft cache hits, we now extend our setup to consider the complete problem with cache overlaps (referred to as "femtocaching" [5]). Note however that we *do* consider user mobility, through variables q_{ij}, unlike some works in this framework that often assume static users. Due to space limitations, we focus on the case of fixed content sizes.

In this scenario, a user $i \in \mathcal{N}$ might be covered by more than one small cells $j \in \mathcal{M}$, i.e. $\sum_j q_{ij} \geq 1, \forall i$. A user is satisfied, if she receives the requested content k or *any* other related content (that she will accept), from *any* of the SCs within range. Hence, similarly to Eq. (3), the total cache hit rate SCHR (that includes regular and soft cache hits) can be written as

$$SCHR(i,k,\mathbf{U}) = 1 - \prod_{j=1}^{M}\prod_{n=1}^{K}\left(1 - u_{kn}^i \cdot x_{nj} \cdot q_{ij}\right) \quad (12)$$

since for a cache hit a user i needs to be in the range of a SC j (term q_{ij}) that stores the content n (term x_{nj}), and accept the recommended content (term u_{kn}^i).

Considering (i) the request probabilities p_k^i, (ii) every user in the system, and (iii) the capacity constraint, gives us the following optimization problem.

Optimization Problem 3. *The optimal cache placement problem for the femtocaching scenario with soft cache hits and content relations described by* $\mathbf{U} = \{u_{kn}\}$ *is*

$$\underset{X=\{x_{11},\ldots,x_{KM}\}}{\textit{maximize}} f(X) = \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \left(1 - \prod_{j=1}^{M}\prod_{n=1}^{K}\left(1 - u_{kn}^i \cdot x_{nj} \cdot q_{ij}\right)\right), \quad (13)$$

$$\sum_{k=1}^{K} x_{kj} \leq C, \quad \forall j \in \mathcal{M}. \quad (14)$$

The following lemma states the complexity of the above optimization problem, as well as its characteristics that allow us to design an efficient approximation algorithm.

Lemma 4.
(1) The Optimization Problem 3 is NP-hard,
(2) with submodular and monotone objective function (Eq. (13)*) and a matroid constraint (Eq.* (14)*).*

Proof. The proof is given in Appendix E. □

Lemma 4 states that the Optimization Problem 3 is a maximization problem with a submodular function and a matroid constraint. For this type of problems, a greedy algorithm can guarantee an $\frac{1}{2}$-approximation of the optimal solution [24]. We propose such a greedy algorithm in Algorithm 3, which is of computational complexity $O\left(K^2M^2\right)$.

Theorem 3. *Let* OPT *be the optimal solution of the Optimization Problem 3, and* S^* *the output of Algorithm 3. Then, it holds that*

$$f(S^*) \geq \frac{1}{2} \cdot OPT \quad (15)$$

Submodular optimization problems have received considerable attention recently, and a number of sophisticated approximation algorithms have been considered (see, e.g., [24] for a survey). For example, a better $\left(1 - \frac{1}{e}\right)$-approximation can be found following the "multilinear extension" approach [26], based on a continuous relaxation and pipage rounding. A similar approach has also been followed in the original femtocaching paper [5]. Other methods also exist that can give an $\left(1 - \frac{1}{e}\right)$-approximation [27]. Nevertheless, minimizing algorithmic complexity or optimal approximation algorithms are beyond the scope of this paper. Our goal instead is to derive fast and efficient algorithms (like greedy) that can handle the large content catalogues and content related graphs $\mathbf{U}$, and compare the performance improvement offered by soft cache hits. The worst-case performance guarantees offered by these algorithms are added value.

V. Femtocaching with Related Content Delivery

We have so far considered a system corresponding to the example of Fig 1(a), where a cache-aware system *recommends* alternative contents to users (in case of a cache miss), but users might not accept them. In this section, we consider a system

Algorithm 3 Greedy Algorithm for Optimization Problem 3. *computation complexity:* $O\left(K^2 \cdot M^2\right)$

Input: *utility* $\{u^i_{kn}\}$, *content demand* $\{p^i_k\}$, *mobility* $\{q_{ij}\}$, $\forall k, n \in \mathcal{K},\ i \in \mathcal{N},\ j \in \mathcal{M}$

$A \leftarrow \mathcal{K} \times \mathcal{M};\ S_0 \leftarrow \emptyset;\ t \leftarrow 0$
for $j \in \mathcal{M}$ **do**
  $c_j \leftarrow 0$
end for
while $A \neq \emptyset$ **do**
  $t \leftarrow t + 1$
  $(n, j) \leftarrow \underset{(k,\ell)\in A}{argmax}\ f(S_{t-1} \cup \{(k, \ell)\})$
  ▷ where, n: content; j: cache/SC
  if $c_j + 1 \leq C$ **then**
    $c_j \leftarrow c_j + 1$
    $S_t \leftarrow S_{t-1} \cup \{(n, j)\}$
  else
    $S_t \leftarrow S_{t-1} \cup \{(n, j)\}$,
  end if
  $A \leftarrow A \backslash \{(n, j)\}$
end while
$S^* \leftarrow S_t$
return S^*

closer to our second example of Fig 1(b), where the system *delivers* some related content that is locally available *instead of the original content*, in case of a cache miss. While a more extreme scenario, we believe this might still have application in a number of scenarios, as explained in Section I (e.g., for low rate plan users under congestion, or in limited access scenarios [15], [16]). We are therefore interested to model and analyze such systems as well. Due to space limitation, we present only the more generic femto-cache case of Section IV; the analysis and results for the single cache cases of Section III follow similarly.

Since original requests might not be served, the (soft) cache hit ratio metric does not describe sufficiently the performance of this system. To this end, we modify the definition of content utility:

Definition 2. *When a user i requests a content k that is not locally available and the content provider delivers an alternative content n then the user satisfaction is given by the utility u^i_{kn}. $u^i_{kn} \in R$ is a real number, and does not denote a probability of acceptance, but rather the happiness of user i when she receives n instead of k. Furthermore $u^i_{kk} = U_{max}, \forall i$.*

Note: we stress that the utilities u^i_{kn} in Def. 2 do not represent the probability a user i to accept a content n (as in Def. 1), but the satisfaction of user i given that she accepted content n. User satisfaction can be estimated by past statistics, or user feedback, e.g., by asking user to rate the received alternative content.

Let us denote as $G_i(t) \subseteq \mathcal{M}$ the set of SCs with which the user i is associated at time t. Given Def. 2, when a user i requests at time t a content k that is not locally available, we assume a system (as in Fig. 1(b)) that delivers to the user the cached content with the *highest* utility[3], i.e., the content n where

$$n \equiv \arg\max_{\ell \in \mathcal{K}, j \in G_i(t)} \left\{u^i_{k\ell} \cdot x_{\ell j}\right\} \tag{16}$$

Hence, the satisfaction of a user i upon a request for content k is

$$\max_{n \in \mathcal{K}, j \in G_i(t)} \left\{u^i_{kn} \cdot x_{nj}\right\} \tag{17}$$

Using the above expression and proceeding similarly to Section IV, it easily follows that the optimization problem that the network needs to solve to optimize the total user satisfaction in the system (among all users and all content requests), which we call *soft cache hit user satisfaction* (SCH-US), is:

Optimization Problem 4. *The optimal cache placement problem for the femtocaching scenario with alternative content delivery and content relations described by the matrix* $\mathbf{U} = \{u^i_{kn}\}$, *is*

$$\underset{X=\{x_1,\ldots,x_K\}}{\textit{maximize}}\ f(X) = \sum_{i=1}^{N}\sum_{k=1}^{K} p^i_k \cdot \max_{n \in \mathcal{K}, j \in \mathcal{M}} \left(u^i_{kn} \cdot x_{nj} \cdot q_{ij}\right), \tag{18}$$

$$\sum_{k=1}^{K} x_{kj} \leq C, \quad \forall j \in \mathcal{M}. \tag{19}$$

For the sub-cases 1 and 2 of Def. 1 presented in Section II-B, the following corollary holds.

Corollary 2. *The expression of Eq.* (18) *needs to be modified as*

$$\max_{n \in \mathcal{K}, j \in \mathcal{M}} \left(u^i_{kn} \cdot x_{nj} \cdot q_{ij}\right) \rightarrow q_{ij} \cdot E\left[\max_{n \in S}\{u^i_{kn}\}\right] = \\ = q_{ij} \cdot \int \left(1 - \prod_{n \in S} F_{kn}(x)\right) dx \tag{20}$$

$$u^i_{kn} \rightarrow u_{kn} \tag{21}$$

(where $S = \{(\ell, m) : \ell \in \mathcal{K}, m \in \mathcal{M}, x_{\ell m} = 1\}$) for the sub-cases 1 and 2 of Def. 1, respectively.

Proof. The proof is given in Appendix F. □

We now prove the following Lemma, which shows that Theorem 1 and Algorithm 1 apply also to the Optimization Problem 4.

Lemma 5.
(1) The Optimization Problem 4 is NP-hard,
(2) with submodular and monotone objective function (Eq. (18)*).*

Proof. The proof is given in Appendix G. □

[3]Equivalently, the system can recommend all the stored contents to the user and then allow the user to select the content that satisfies her more.

VI. Performance Evaluation

A. Simulation setup

Contents dataset. We consider a real dataset of YouTube videos from [28]. The dataset contains several information about the videos, such as their (a) popularity and (b) size, as well as (c) the list of related videos (as recommended by YouTube) for each of them. This information allows us to simulate scenarios with real parameters p_k^i, s_k, and u_{kn}^i. After pre-processing the data to remove contents with no popularity values, we build the related content matrix (utility matrix $\mathbf{U}$). Due to the sparsity of the dataset, we only consider contents belonging to the largest connected component, that includes $K = 2098$ videos. The average number of related content for these videos is 3.6. Since our dataset does not contain per-user information, we consider the sub-case-2 of Definition 1, and assume that if content k is related to content n in the dataset, then $u_{kn} = 1$. However, we later perform a sensitivity analysis as a function of diminishing acceptance probabilities for related content.

Cellular network. We consider an area of 1 km^2 that contains $M = 20$ SCs. SCs are randomly placed (i.e., uniformly) in the area, which is an assumption that has been also used in similar works [5], [29]. An SC can serve a request from a user, when the user is inside its communication range, which we set to 200 meters. We also consider $N = 50$ mobile users. This creates a relatively dense network, where a random user is connected to 3 SCs *on average*. We will also consider sparser and denser scenarios, for comparison. We generate a set of 20 000 requests according to the content popularity calculated from the UouTube dataset, over which we average our results.

Unless otherwise stated, the simulations use the parameters summarized in Table II.

TABLE II: Parameters used in the simulations.

Parameter	Value	Parameter	Value
nb. of contents, K	2098	nb. of requests	20.000
Cache size, C	5	nb. of SCs, M	20
Area	1 x 1 km	Communication range	200 m

B. Performance Results

We consider the following four content caching schemes:

- *Single (popularity-based)*: Single cache accessible per user (e.g., the closest one). Only normal cache hits allowed, and the most popular contents are stored in each cache. This is the baseline scheme, commonly used in related works.
- *SingleSCH*: Single cache with soft cache hits, with the content allocation given by Algorithm 1 (or Algorithm 2).
- *Femto*: Femto-caching without soft cache hits (from [5]).
- *FemtoSCH*: Femto-caching with soft cache hits, with the content allocation given by Algorithm 3.

Cache size impact: We first investigate the impact of cache size, assuming fixed content sizes. Fig. 2 depicts the total cache hit ratio, for different cache sizes C: we consider a cache size per SC between 2 and 15 contents. The simulations suggest that soft cache hits (SCH) can double the cache hit ratio. What is more, these gains are applicable to both the single cache and femto-caching scenarios, which show that our approach can offer considerable benefits *on top of femto-caching*, which as we see can already achieve an improvement of more than 50%, compared to single caches in this scenario. The two methods together offer a total of $3\times$ improvement compared to the baseline scenario "Single", reaching a maximum cache hit ratio of about 60% for $C = 15$. Finally, even with a cache size of per SC of about 0.1% of the total catalog, introducing soft cache hits offers 30% cache hit ratio, which is promising.

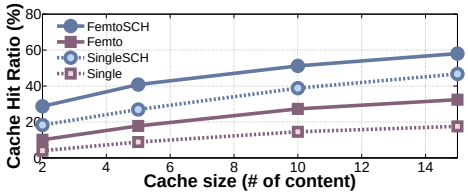


Fig. 2: Cache hit ratio vs. C, with fixed content size.

Variable file size: In Fig. 3, we now also take into account the different file sizes when optimizing our allocation (these are available in the YouTube dataset). Comparing the performance of even this less theoretically efficient greedy algorithm (Algorithm 2) to the single cache with no soft hits, already reveals considerable performance improvement. In fact, Algorithm 2 exploits the fact that contents have different sizes, to "replace" longer contents with related ones that might be shorter. While one could of course not replace a very large content with a very small one, we have observed in our dataset that the sizes of related contents are not independent (i.e., related videos of large videos are indeed large, and vice versa, but have enough difference that can sometimes be exploited by the size-aware algorithm).

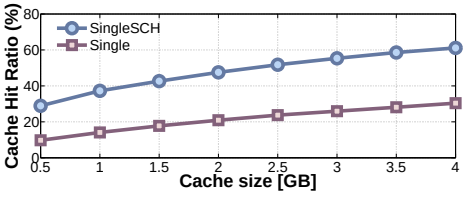


Fig. 3: Cache hit ratio vs. C with variable content size.

SC density impact: In Fig. 4 we perform a sensitivity analysis with respect to the number of SCs in the area (assuming fixed capacity $C = 5$). We test 2 sparse scenarios ($M = 5$ and $M = 10$) and 2 dense scenarios ($M = 20$ and $M = 30$). The average number of SCs that can be seen by a user varies from around 1 ($M = 5$) to 4.6 ($M = 30$). In the sparse scenarios, a user can usually see at most one SC. For this reason, "Femto" and "Single" perform similarly ($20-30\%$ cache hit rate). As the SC density increases, the basic femto-caching is able to improve performance, as expected. However, femtocaching with SCH brings even more performance gains.

With a storage capacity per SC of about 0.25% of the content catalog (5 out of 2000 contents), and a coverage overlap of 2-4 SCs per user, femto-caching together with SCH can achieve a $30-50\%$ cache hit ratio. This is promising on the additive gains of the two methodologies.

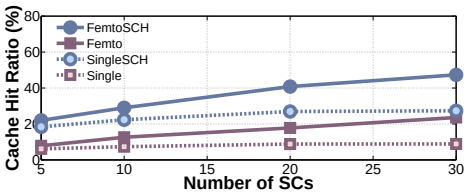


Fig. 4: Cache hit ratio vs. number of SCs M ($C = 5$).

Utility matrix impact: In these final two sets of simulations, we further investigate the impact of the content relations as captured by the matrix $\mathbf{U}$ (and its structure). A first important parameter to consider is the average number of related contents per video, which we denote as $E[R]$, where the set R for a content k is defined as $R = \{n \in \mathcal{K} : u_{kn} > 0\}$. In the previous scenarios, the number of related contents was inferred from the YouTube trace, and was found to be equal to $E[R] = 3.6$. To understand the impact of this parameter, in this next scenario we generate two synthetic content graphs $\mathbf{U}$:

- *SCH(1)*: content k picks content n as related content with probability p_n (i.e., proportional to its popularity), normalized to a mean $E[R]$ value per content.
- *SCH(2)*: any content picks $E[R]$ related content randomly chosen.

While the latter assumes that content relations are independent of their popularity, the former assumes that a more popular content has a higher chance to appear in the related content list. In fact, this is quite inline with daily experience of how recommendation systems work.

In Fig. 5, we compare the cache hit ratio for single and femto-caching scenarios: without SCH, with SCH1, and with SCH2, assuming that $\mathbf{E}[R]$ varies between 2 and 10 related content items. A first observation is that, due to the sparsity of the content matrix, SCH(2) (i.e., random content relations) brings only marginal improvements to the total number of hits. On the other hand, a correlation between related content and popularity (i.e., SCH(1)) is what brings considerable offloading gains, even for small $E[R]$. In fact, comparing these synthetic results with the previous trace-based ones, one can infer that the real dataset probably more closely resembles SCH1, i.e. does exhibit such a correlation.

User flexibility: In this last scenario, we present in Fig.6 the cache hit ratio as a function of the willingness of a user to accept a related content. We consider two scenarios, both in the femto-caching context:

- *Synthetic:* We generate a synthetic matrix U as in SCH(1) above, with $E[R] = 4$ and $u_{kn} = u < 1$.
- *YouTube:* We use the real YouTube dataset for the matrix U. However, all related contents also have a utility $u_{kn} = u < 1$ (instead of $u_{kn} = 1$ considered in the previous scenarios).

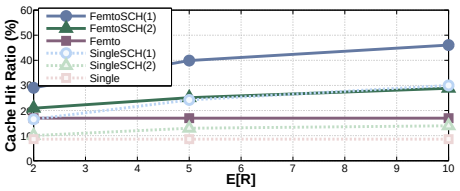


Fig. 5: Cache hit ratio for different number of related contents $E[R]$; synthetic traces.

On the x-axis of Fig.6 we vary this parameter u from 0 to 1. As can be seen there, when the user's acceptance probability becomes very small, the scenario becomes equivalent to standard femto-caching without soft hits, and the gains reported there are inline with the previous plots. However, as user willingness to accept related content increases, the optimization policy can exploit opportunities for potential soft cache hits and improve performance. E.g. for a probability 50% to accept an alternative recommended content, cache hit ratios increase by almost $2\times$ (from 15% to 27% in the YouTube dataset). Results are in fact very comparable for the synthetic and YouTube traces. Finally, we observed similar behavior in scenarios conforming to the model of Section V, where u_{kn} do not denote probabilities of acceptance, but correspond to the user satisfaction.

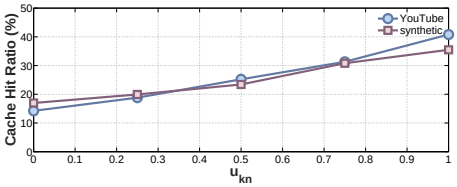


Fig. 6: Cache hit ratio as a function of user's willingness to accept related content ($M = 20, C = 5, E[R] = 4$).

VII. Related Work

Mobile Edge Caching. Densification of cellular networks, overlaying the standard macro-cell network with a large number of small cells (e.g., pico- or femto-cells), has been extensively studied and is considered a promising solution to cope with data demand [30], [31], [32]. As this densification puts a tremendous pressure on the backhaul network, researchers have suggested storing popular content at the "edge", e.g., at small cells [5], user devices [33], [9], [8], or vehicles acting as mobile relays [11].

Our work is complementary to these approaches, as it can utilize such mobile edge caching systems while showing how to further optimize the cache allocation when there is a cache-aware recommender systems in place. We have applied this approach in the context of mobile (ad-hoc) networks with delayed content delivery [19] as well, and applied it here for the first time in the context of femto-caching [5]. Additional

research directions have also recently emerged, more closely considering the interplay between caching and the physical layer such as Coded Caching [7] and caching for coordinated (CoMP) transmission [34], [35]. We believe the idea of soft cache hits could be applied in these settings as well, and we plan to explore this as future work.

Caching and Recommendation Interplay. Although not in the area of wireless systems, there exist some recent works that have jointly considered caching and recommendation in peer-to-peer networks [36] and CDNs [37], [38]. Specifically, [36] studies the interplay between a recommendation system and the performance of content distribution on a peer-to-peer network like BitTorrent. The authors model and analyse the performance, and propose heuristics (e.g., based on the number of cached copies, or "seeders" in BitTorrent) for tuning the recommendation system, in order to improve performance and reduce content distribution costs. Although in our work we follow the opposite approach (given a recommendation system and user utilities, we optimize the caching algorithm), it would be interesting to investigate such recommendation system optimizations in the context of cellular networks.

[37] shows that users tend to follow YouTube's suggestions, and despite the large catalog of YouTube, the top-10 recommendations are usually common for different users in the same geographical region. Hence, CDNs can use the knowledge from the recommendation system to improve their content delivery (e.g., by storing in central caches the most appropriate contents). Finally, in [38] the authors suggest a recommended list reordering approach that can achieve higher cache hit ratios (for YouTube servers/caches). In particular, they show that performance can be improved when the positions of contents in the related list of YouTube are not changed compared to the previous recommended list. The increasing dependence of user requests on the output of recommender systems clearly suggests that there is an opportunity to further improve the performance of (mobile) edge caching by jointly optimizing both, with minimum impact on user QoE.

VIII. Conclusions

In this paper, we have proposed the idea of *soft cache hits*, where an alternative content can be recommended to a user, when the one she requested is not available in the local cache. While normal caching systems would declare a cache miss in that case, we argue that an appropriate recommended content, related to the original one can still satisfy the user with high enough probability. We then used this idea to design such a system around femto-caching, and demonstrated that considerable additional gains, *on top of those of femto-caching* can be achieved using realistic scenarios and data. We believe this concept of soft cache hits has wider applicability in various caching systems.

References

[1] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, "Five Disruptive Technology Directions for 5G," *IEEE Comm. Mag. SI on 5G Prospects and Challenges*, 2014.

[2] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, pp. 1–9, 2010.

[3] G. S. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *http://arxiv.org/abs/1602.00173*, 2016.

[4] G. Paschos and P. Elia, "Caching for wireless networks," in *IEEE Sigmetrics Tutorials*, 2016.

[5] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, 2012.

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

[7] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, pp. 2856–2867, May 2014.

[8] P. Sermpezis and T. Spyropoulos, "Not all content is created equal: Effect of popularity and availability for content-centric opportunistic networking," in *Proc. ACM MobiHoc*, pp. 103–112, 2014.

[9] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Trans. on Mobile Computing*, vol. 11, no. 5, 2012.

[10] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan, "Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays," in *Proc. IEEE WoWMoM*, 2011.

[11] L. Vigneri, T. Spyropoulos, and C. Barakat, "Storage on Wheels: Offloading Popular Contents Through a Vehicular Cloud," in *Proc. IEEE WoWMoM*, 2016.

[12] "T-Mobile Music Freedom." https://www.t-mobile.com/offer/free-music-streaming.html 2017.

[13] Y. Yiakoumis, S. Katti, and N. McKeown, "Neutral net neutrality," in *Proc. ACM SIGCOMM*, pp. 483–496, 2016.

[14] S. Guo, M. Derakhshani, M. Falaki, U. Ismail, R. Luk, E. Oliver, S. U. Rahman, A. Seth, M. Zaharia, and S. Keshav, "Design and implementation of the kiosknet system," *Computer Networks*, vol. 55, no. 1, pp. 264–281, 2011.

[15] "Internet.org by Facebook." https://info.internet.org/, 2017.

[16] "free basics by Facebook." https://0.freebasics.com/desktop, 2017.

[17] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358–380, 2015.

[18] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "Cellslice: Cellular wireless resource slicing for active ran sharing," in *COMSNETS*, pp. 1–6, 2014.

[19] T. Spyropoulos and P. Sermpezis, "Soft cache hits and the impact of alternative content recommendations on mobile edge caching," in *Proc. ACM Workshop on Challenged Networks (CHANTS)*, pp. 51–56, 2016.

[20] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, pp. 4:2–4:2, 2009.

[21] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[22] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.

[23] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Mathematics of operations research*, vol. 3, no. 3, pp. 177–188, 1978.

[24] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, no. 19, p. 8, 2012.

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

[26] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.

[27] Y. Filmus and J. Ward, "Monotone submodular maximization over a matroid via non-oblivious local search," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 514–542, 2014.

[28] http://netsg.cs.sfu.ca/youtubedata/, 2007.

[29] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *Proc. IEEE INFOCOM*, pp. 1078–1086, 2014.

[30] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan, "Heterogeneous cellular networks: From theory to practice," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 54–64, 2012.
[31] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 497–508, 2012.
[32] J. G. Andrews, "Seven ways that hetnets are a cellular paradigm shift," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 136–144, 2013.
[33] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," *http://arxiv.org/abs/1205.7044*, 2012.
[34] W. C. Ao and K. Psounis, "Distributed caching and small cell cooperation for fast content delivery," in *Proc. ACM MobiHoc*, 2015.
[35] A. Liu and V. K. Lau, "Cache-enabled opportunistic cooperative mimo for video streaming in wireless systems," *IEEE Trans. Signal Processing*, 2015.
[36] D. Munaro, C. Delgado, and D. S. Menasché, "Content recommendation and service costs in swarming systems," in *Proc. IEEE ICC*, pp. 5878–5883, 2015.
[37] D. K. Krishnappa, M. Zink, and C. Griwodz, "What should you cache?: a global analysis on youtube related video caching," in *Procc. ACM NOSSDAV Workshop*, pp. 31–36, 2013.
[38] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-centric video recommendation: an approach to improve the efficiency of youtube caches," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 4, p. 48, 2015.
[39] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.
[40] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. ACM SIGKDD*, pp. 420–429, 2007.

APPENDIX A
PROOF OF COROLLARY 1

Sub-case 1. Since the exact per-user utilities are not known, we calculate the SCHR given in Eq. (2) by taking the conditional expectations on $F_{kn}(x)$. Denoting the corresponding pdf as $f_{kn}(x)$, we proceed as follows:

$$\begin{aligned}
SCHR &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot E\left[\left(1-\prod_{n=1}^{K}\left(1-u_{kn}^i \cdot x_n\right)\right)\right] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1-E\left[\prod_{n=1}^{K}\left(1-u_{kn}^i \cdot x_n\right)\right]\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1-\prod_{n=1}^{K} E\left[\left(1-u_{kn}^i \cdot x_n\right)\right]\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1-\prod_{n=1}^{K} \int (1-t\cdot x_n)\cdot f_{kn}(t)dt\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1-\prod_{n=1}^{K} \left(1-\left(\int t\cdot f_{kn}(t)dt\right)\cdot x_n\right)\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1-\prod_{n=1}^{K} \left(1-E[u_{kn}^i]\cdot x_n\right)\right)
\end{aligned}$$

where (i) the third equation holds since the utilities for different content pairs {k,n} are independent, and thus the expectation of their product is equal to the product of their expectations, and (ii) we denoted

$$E[u_{kn}^i] \equiv \int t\cdot f_{kn}(t)dt = \int (1-F_{kn}(t))dt$$

and the above equation holds since u_{kn}^i is a positive random variable.

Sub-case 2 follows straightforwardly.

APPENDIX B
PROOF OF LEMMA 2

We prove here the NP-hardness of the optimal cache allocation for a single cache with soft cache hits. Let us consider an instance of Optimization Problem 1, where the utilities are equal among all users and can be either 1 or 0, i.e., $u_{kn}^i = u_{kn},\ \forall i \in \mathcal{N}$ and $u_{kn} \in \{0,1\},\ \forall k,n \in \mathcal{K}$. We denote as $\mathcal{R}_k$ the set of contents related to content k, i.e.

$$\mathcal{R}_k = \{n \in \mathcal{K} : n \neq k, u_{kn} > 0\} \text{ (related content set)} \quad (22)$$

Consider the content subsets $\mathcal{S}_k = \{k\} \cup \mathcal{R}_k$. Assume that only content k is stored in the cache ($x_k = 1$ and $x_n = 0, \forall n \neq k$). All requests for contents in S_k will be satisfied (i.e. "covered" by content k), and thus SCHR will be equal to $\sum_{i\in\mathcal{N}}\sum_{n\in S_k} p_n^i \cdot q_i$. When more than one contents are stored in the cache, let $\mathcal{S}^{'}$ denote the union of all contents covered by the stored ones, i.e., $\mathcal{S}^{'} = \bigcup_{\{k:x_k=1\}} S_k$. Then, the SCHR will be equal to $\sum_{i\in\mathcal{N}}\sum_{n\in S^{'}} p_n^i \cdot q_i$. Hence, the Optimization Problem 1 becomes equivalent to

$$\max_{\mathcal{S}^{'}} \quad \sum_{n\in\mathcal{S}^{'}} p_n^i \cdot q_i \qquad s.t. \quad |\{k : x_k = 1\}| \leq C.$$

This corresponds to the the *maximum coverage problem with weighted elements*, where "elements" (to be "covered") correspond to the contents $i \in \mathcal{K}$, weights correspond to the probability values $p_n^i \cdot q_i$, the number of selected subsets $\{k : x_k = 1\}$ must be less than C, and their union of covered elements is $\mathcal{S}^{'}$. This problem is known to be a NP-hard problem [39], and thus the more generic problem (with different u_{kn}^i and $0 \leq u_{kn} \leq 1$) is also NP-hard.

APPENDIX C
PROOF OF LEMMA 3

The objective function of Eq. (6) $f(X) : \{0,1\}^K \to \mathbb{R}$ is equivalent to a set function $f(S) : 2^{\mathcal{K}} \to \mathbb{R}$, where $\mathcal{K}$ is the finite *ground set* of contents, and $S = \{k \in \mathcal{K} : x_k = 1\}$. In other words,

$$f(S) \equiv \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1-\prod_{n\in S}\left(1-u_{kn}^i\right)\right). \quad (23)$$

A set function is characterised as *submodular* if and only if for every $A \subseteq B \subset V$ and $\ell \in V \backslash B$ it holds that

$$[f(A\cup\{\ell\}) - f(A)] - [f(B\cup\{\ell\}) - f(B)] \geq 0 \quad (24)$$

From Eq. (6), we first calculate

$$
\begin{aligned}
&f\left(A\cup\{\ell\}\right)-f\left(A\right)=\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\left(1-\prod_{n\in A\cup\{\ell\}}\left(1-u_{kn}^i\right)\right)\\
&\qquad-\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\left(1-\prod_{n\in A}\left(1-u_{kn}^i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\cdot q_i\cdot\left(\prod_{n\in A}\left(1-u_{kn}^i\right)-\prod_{n\in A\cup\{\ell\}}\left(1-u_{kn}^i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\cdot q_i\cdot\left(u_{k\ell}^i\cdot\prod_{n\in A}\left(1-u_{kn}^i\right)\right).
\end{aligned}
$$

Then,

$$
\begin{aligned}
&\left[f\left(A\cup\{\ell\}\right)-f\left(A\right)\right]-\left[f\left(B\cup\{\ell\}\right)-f\left(B\right)\right]=\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\left(u_{k\ell}^i\prod_{n\in A}\left(1-u_{kn}^i\right)\right)\\
&\qquad-\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\left(u_{k\ell}^i\prod_{n\in B}\left(1-u_{kn}^i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\cdot u_{k\ell}^i\cdot\left(\prod_{n\in A}\left(1-u_{kn}^i\right)-\prod_{n\in B}\left(1-u_{kn}^i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\cdot u_{k\ell}^i\cdot\prod_{n\in A}\left(1-u_{kn}^i\right)\cdot\left(1-\prod_{n\in B\setminus A}\left(1-u_{kn}^i\right)\right)
\end{aligned}
$$

The above expression is always ≥ 0, which proves the submodularity for function f.

Furthermore, the function f is characterised as *monotone* if and only if $f(B)\geq f(A)$ for every $A\subseteq B\subset V$. In our case, this property is shown as

$$
\begin{aligned}
&f(B)-f(A)=\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\cdot\left(1-\prod_{n\in B}\left(1-u_{kn}^i\right)\right)\\
&\qquad-\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\cdot\left(1-\prod_{n\in A}\left(1-u_{kn}^i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\cdot\left(\prod_{n\in A}\left(1-u_{kn}^i\right)-\prod_{n\in B}\left(1-u_{kn}^i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i q_i\cdot\prod_{n\in A}\left(1-u_{kn}^i\right)\cdot\left(1-\prod_{n\in B\setminus A}\left(1-u_{kn}^i\right)\right)\geq 0
\end{aligned}
$$

Appendix D
Proof of Theorem 2

Following similar arguments as in the proof of Lemma 2, the Optimization Problem 2 can be shown to be equivalent to the *budgeted maximum coverage problem with weighted elements*, which is an NP-hard problem [39].

In Algorithm 2, we first calculate a solution $S^{(1)}$ returned by a modified version (ModifiedGreedy) of the greedy algorithm (line 1). The differences between the greedy algorithm (e.g., Algorithm 1) and ModifiedGreedy, are that the latter: (a) each time selects to add in the cache the content that increases the most the fraction of the objective function over its own size (line 13), and (b) considers every content, until there is no content that can fit in the cache (lines 14-20). Then, Algorithm 2 calculates the solution $S^{(2)}$ that the greedy algorithm would return if all contents were of equal size (line 2). The returned solution, is the one between $S^{(1)}$ and $S^{(2)}$ that achieves a higher value of the objective function (lines 3-7).

Hence, Algorithm 2 is a "fast-greedy" type of approximation algorithm. Since, the objective function was shown to be submodular and monotone in Lemma 3, our fast greedy approximation algorithm can achieve a $\frac{1}{2}\cdot\left(1-\frac{1}{e}\right)$-approximation solution (in the worst case), when there is a Knapsack constraint, using similar arguments as in [40].

Appendix E
Proof of Lemma 4

Item (1): In the single cache case, we reduced the Optimization Problem 1 to a weighted maximum coverage problem, where a set of contents need to be selected (i.e., to be stored) in order to maximize the weights (i.e., probabilities $p_k^i\cdot q_i$) of the "elements" (i.e., other contents) with which they are connected (i.e., edges/utilities $u_{kn}^i>0$). The ground set of contents was $\mathcal{K}$, and the ground set of "elements" was $\mathcal{K}$ as well.

In the case of multiple users and overlapping caches, the Optimization Problem 3 can similarly be reduced to the NP-hard weighted maximum coverage problem, if (i) instead of the set of contents $\mathcal{K}$, we consider the set of tuples {content,user} $\mathcal{K}\times\mathcal{N}$, and (ii) instead of "elements"/contents, we consider the set of "elements"/tuples {content,cache} $\mathcal{K}\times\mathcal{M}$.
Item (2): We consider the objective function of Eq. (13) and apply the same steps as in proof of Lemma 3. Specifically, for all sets $A\subseteq B\subset\mathcal{K}\times\mathcal{M}$ and {content, SC} tuples $(\ell,m)\in V\setminus B$, we get

$$
\begin{aligned}
&f\left(A\cup\{(\ell,m)\}\right)-f\left(A\right)=\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\left(1-\prod_{(n,j)\in A\cup\{(\ell,m)\}}\left(1-u_{kn}^i\cdot q_{ij}\right)\right)\\
&\quad-\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\left(1-\prod_{(n,j)\in A}\left(1-u_{kn}^i\cdot q_{ij}\right)\right)\\
&=\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\cdot u_{k\ell}^i\cdot q_{im}\cdot\prod_{(n,j)\in A}\left(1-u_{kn}^i\cdot q_{ij}\right)\geq 0
\end{aligned}
$$

which proves *submodularity*, and

$$
\begin{aligned}
&f(B)-f(A)=\\
&\quad\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\left(1-\prod_{(n,j)\in B}\left(1-u_{kn}^i q_{ij}\right)\right)-\\
&\qquad\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\left(1-\prod_{(n,j)\in A}\left(1-u_{kn}^i q_{ij}\right)\right)=\\
&\sum_{i=1}^{N}\sum_{k=1}^{K}p_k^i\prod_{(n,j)\in A}\left(1-u_{kn}^i\cdot q_{ij}\right)\cdot\left(1-\prod_{(n,j)\in B\setminus A}\left(1-u_{kn}^i\cdot q_{ij}\right)\right)\geq 0
\end{aligned}
$$

which proves *monotonicity*.

To show that the constraint is a matroid (see e.g. [24] for the definition of a matroid), we consider the set $\mathcal{V} = \mathcal{K} \times \mathcal{M}$ (i.e., all the possible tuples {content,cache}) and the collection of subsets of $\mathcal{V}$ that do not violate the capacity of the caches

$$I = \left\{ S \subseteq 2^{\mathcal{V}} : |S \cap 2^{\{\mathcal{K},m\}}| \leq C, \quad \forall m \in \mathcal{M} \right\}$$

Then:
(a) For all sets A and B that $A \subseteq B \subseteq \mathcal{V}$, it holds that if $B \subseteq \mathcal{I}$ (i.e., the caching placement defined by B does not violate the size of the caches) then $A \subseteq \mathcal{I}$, because in A every cache has to store the same or less content than in B and thus no capacity constraint is violated.
(b) For all sets $A, B \in \mathcal{I}$ (i.e., feasible caching placements) and $|B| > |A|$ (i.e., in B more contents are cached), $\exists \ell \in B \backslash A$ that $A \cup \{\ell\} \in \mathcal{I}$, since in A not all caches are full (otherwise B would violate the capacity constraint, i.e., $B \notin \mathcal{I}$), which means that there exists at least one more content can be cached (and this content can be selected to be from the set B).

From (a) and (b), it follows directly that the constraint is a matroid [24].

Appendix F
Proof of Corollary 2

Sub-case 1. Similar to the proof Corollary 1, by taking the conditional expectations on the different $F_{kn}(x)$ in Eq. (18), we get:

$$SCHR = \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot E\left[\max_{n \in \mathcal{K}, j \in \mathcal{M}} \left(u_{kn}^i \cdot x_{nj} \cdot q_{ij} \right) \right] \quad (25)$$

Let us consider the random variable Y_{kS}, where $Y_{kS} = \max_{(n,j)\in S} \left(u_{kn}^i \right)$, where $S = \{(\ell, m) : \ell \in \mathcal{K}, m \in \mathcal{M}, x_{\ell m} = 1\}$. The distribution of Y_{kS} (as the max value of independent random variables) is given by

$$F_{kS}(x) = P\{Y_{kS} \leq x\} = \prod_{(n,j)\in S} F_{kn}(x) \quad (26)$$

Then Eq. (25) becomes

$$\begin{aligned} SCHR &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot E\left[Y_{kS} \cdot q_{ij} \right] \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot E\left[Y_{kS} \right] \cdot q_{ij} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot \left(\int \left(1 - F_{kS}(t) \right) dt \right) \cdot q_{ij} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot \left(\int \left(1 - \prod_{(n,j)\in S} F_{kn}(t) \right) dt \right) \cdot q_{ij} \end{aligned}$$

Sub-case 2 follows straightforwardly.

Appendix G
Proof of Lemma 5

Item (1): Optimization Problem 4 is of the exact same nature as Optimization Problem 3, so it follows that it is NP-hard.
Item (2): We proceed similarly to the proof of Lemma 3. The objective function of Eq. (18) $f(X) : \{0,1\}^{K \times M} \to \mathbb{R}$ is equivalent to a set function $f(S) : 2^{\mathcal{K} \times \mathcal{M}} \to \mathbb{R}$, where $\mathcal{K}$ and $\mathcal{M}$ are the finite *ground sets* of contents and SCs, respectively, and $S = \{k \in \mathcal{K}, j \in \mathcal{M} : x_{kj} = 1\}$:

$$f(S) \equiv \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \cdot \max_{(n,j)\in S} \left(u_{kn}^i \cdot q_{ij} \right) \quad (27)$$

For all sets $A \subseteq B \subset V$ and {content, SC} tuples $(\ell, m) \in V \backslash B$, we get

$$\begin{aligned} &f\left(A \cup \{(\ell,m)\}\right) - f\left(A\right) = \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \max_{(n,j)\in A\cup\{(\ell,m)\}} \left(u_{kn}^i q_{ij} \right) \\ &\qquad - \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i q_i \max_{(n,j)\in A} \left(u_{kn}^i q_{ij} \right) \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i R\left(u_{k\ell}^i \cdot q_{im} - \max_{(n,j)\in A} \left(u_{kn}^i q_{ij} \right) \right) \end{aligned}$$

where in the last equation we use the *ramp function* defined as $R(x) = x$ for $x \geq 0$ and $R(x) = 0$ for $x < 0$. Subsequently,

$$\begin{aligned} &[f\left(A \cup \{(\ell,m)\}\right) - f\left(A\right)] - [f\left(B \cup \{(\ell,m)\}\right) - f\left(B\right)] = \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \left[R\left(u_{k\ell}^i q_{im} - \max_{(n,j)\in A} \left(u_{kn}^i q_{ij} \right) \right) \right. \\ &\qquad \left. - R\left(u_{k\ell}^i q_{im} - \max_{(n,j)\in B} \left(u_{kn}^i q_{ij} \right) \right) \right] \end{aligned}$$

The above equation is always ≥ 0 (which proves that the objective function Eq. (18) is *submodular*), since the ramp function is monotonically increasing and comparing the two arguments of the function $R(x)$ in the above equation, gives

$$\begin{aligned} &u_{k\ell}^i q_{im} - \max_{(n,j)\in A} \left(u_{kn}^i q_{ij} \right) - \left(u_{k\ell}^i q_{im} - \max_{(n,j)\in B} \left(u_{kn}^i q_{ij} \right) \right) = \\ &\qquad \max_{(n,j)\in B} \left(u_{kn}^i q_{ij} \right) - \max_{(n,j)\in A} \left(u_{kn}^i q_{ij} \right) \geq 0 \end{aligned}$$

since B is a superset of A and therefore its maximum will be at least equal or greater than the maximum value in set A.
Similarly, since $A \subseteq B$ it holds

$$f(B) - f(A) = \sum_{i=1}^{N} \sum_{k=1}^{K} p_k^i \left(\max_{(n,j)\in B} \left(u_{kn}^i q_{ij} \right) - \max_{(n,j)\in A} \left(u_{kn}^i q_{ij} \right) \right) \geq 0$$

which proves that the Eq. (18) is *monotone*.

APPENDIX H
(1-1/*e*) APPROXIMATION ALGORITHM FOR OPTIMIZATION PROBLEM 2

Algorithm 4 $\left(1-\frac{1}{e}\right)$-approximation Algorithm for Optimization Problem 2.

$A \leftarrow \left\{S \subseteq \mathcal{K}:\ |S|<3 \text{ and } \sum_{i\in S} s_i \leq C\right\}$
$S^{(1)} \leftarrow \underset{S\in A}{argmax} f(S)$
$B \leftarrow \left\{S \subseteq \mathcal{K}:\ |S|=3 \text{ and } \sum_{i\in S} s_i \leq C\right\}$
$S^{(2)} \leftarrow \emptyset$
for $S \in B$ **do**
    $U \leftarrow \mathcal{K} \backslash S$
    $W \leftarrow \text{list}(w_i),\ \forall i \in U$
    $H \leftarrow$MODIFIEDGREEDY(U,[W])
    if $f(H) > f(S^{(2)})$ **then**
        $S^{(2)} \leftarrow H$
    end if
end for

if $f(S^{(1)}) > f(S^{(2)})$ **then**
    $S^* \leftarrow S^{(1)}$
else
    $S^* \leftarrow S^{(2)}$
end if
return S^*