

# Content Preference-aware User Association and Caching in Cellular Networks

George Darzanos, Livia Elena Chatzieleftheriou, Merkourios Karaliopoulos, Iordanis Koutsopoulos  
Athens University of Economics and Business, AUEB, Athens, Greece  
{ntarzanos, liviachatzi, mkaralio, jordan}@aueb.gr

**Abstract**—The ever-growing trend of mobile users to consume high quality videos in their devices has pushed the backhaul network to its limits. Caching at Small-cell Base Stations (SBSs) has been established as an effective mechanism for alleviating this burden. Next generation mobile networks promise high Access Network density, hence multiple SBS association options per mobile user may exist. In this paper, we study the joint problem of mobile user association to SBSs and content placement to SBS-collocated caches, aiming to further optimize the utilization of backhaul network. The problem is solved periodically, considering time intervals where the users’ location to the system is assumed to be fixed. First, we decompose the joint problem into two sub-problems that are solve sequentially, namely the content preference similarity-driven user association and the demand aware content placement sub-problems. We then propose a heuristic that consists of multiple phases. In particular, at a preparation phase, it performs clustering of users based on the similarity of their content preferences, accounting also for geographical constraints. The resulting clusters are then utilized for the demand-aware association of users to the SBS, while the placement of content is driven by the resulting local demand in each SBS, and takes place at the end. We demonstrate the effectiveness of our heuristic by evaluating its performance against multiple schemes that either lack a preparation phase or do not account for geographical constraints. As it is evident through the numerical results, the user clustering that takes place during the preparation phase can increase the overall cache hit ratio up to 20%.

**Index Terms**—Clustering, User Association, Caching.

## I. INTRODUCTION

Video on Demand (VoD) traffic toward mobile devices increases exponentially and is going to explode in the next few years. This mainly happens due to the increased mobile users’ daily consumption of video content for multiple purposes, from business to entertainment. According to Cisco [1], there will be a seven-fold increase of the global mobile traffic between 2017 and 2022, with 79% of it being video. In order to mitigate the impact of this trend to the Mobile Network Operators’ (MNOs) backhaul network and maintain high Quality of Experience (QoE) for mobile video viewers, caching at Small-cell Base Station (SBS) has been proposed as effective mechanism [2], [3]. The storage capability of the SBSs is limited, hence they can only maintain a small fraction of content items that mobile users may request through Content Provider (CP) platforms such as Netflix, Amazon Prime Video or YouTube. To this end, sophisticated mechanisms that capture the spacial locality of content demand are needed in order to better utilize this storage space.

Next generation mobile networks support dual connectivity for mobile devices, with simultaneous connectivity to a Macro-Cell Base Station (MBS) and an SBS. Furthermore, the high density of heterogeneous small-cell networks, results in multiple SBSs being in the proximity of a mobile user, giving the MNO a variety of association options. Consequently, user association becomes critical since placing users with similar demand patterns to the same SBS will boost the impact of caching. The joint control over user association and caching can be performed by the MNO with information gathered by the CPs, e.g. user content preferences. The question of how to exploit user similarity in terms of content preferences in order to perform joint optimization is the topic of this paper.

The popularity of content items across different geographical areas is determined by local mobile users’ demand patterns. Given that VoD platforms define multiple thematic categories in order to better classify their content, demand patterns can be extracted by taking advantage of mobile users’ preferences over the different thematic categories. The common practice for users’ association to SBSs is to associate each mobile user to the closest (proximity-wise) SBS. However, this approach does not take into account the user demand patterns, neither does it take advantage of the multiple association options per user. On the other hand, clustering of mobile users based on the similarity of their content preferences, may bring significant benefits for caching. In particular, an association strategy that matches each user cluster to an SBS may increase the cache hit ratio [4]. Recent studies assume that user demand patterns are a priori known [2], [5] or are obtained through Machine Learning techniques [6].

Traditionally, the actions of content placement and user association are performed in two discrete steps. Content placement is mostly driven by content items’ popularity, while user association is usually performed based on user-SBS proximity criteria, ignoring the user demand patterns. In practice, the frequency of cache updates varies from “once a day” to “every few minutes”, depending both on the type of content and the strategy adopted by the cache operator. On the other hand, user association updates are usually triggered by the mobility of users. According to recent literature [7], users follow certain mobility patterns in their daily life and spend most of their time at place of “high-interest”, such as home and work. Consequently, we can consider certain time intervals of few hours where the users are supposed to remain *static*, i.e. in fixed locations. The duration of these time intervals is determined

by events of major shift on users' geographic location. Such an event will trigger a notable number of association updates and a "bulk" update on caches' contents since the demand in each location will be significantly affected.

In this paper, we study the joint problem of associating users to SBSs and caching content to the SBS-located caches, taking into account the mobile users' geographical constraints and the similarity of their content preferences. We decompose the joint problem into two sub-problems that are solved sequentially. User association will take place first and content placement will follow, hence caching decisions are aware of the local demand generated in each SBS. However, it is challenging to a priori decide user association, so that the effectiveness of content placement is boosted. We thus introduce a *preparation phase*, where clusters of mobile users are generated taking into account their content preferences. Associating users from the same cluster to an SBS will indirectly increase the efficiency of caching in terms of cache hit ratio. Our contribution is summarized as follows:

- We formulate the joint problem of user association and caching, which is proven NP-hard. Then, we decompose it into two sub-problems that are sequentially solved, i.e. (i) the *user association sub-problem*, which is driven by the similarity of users' content preferences and (ii) the *caching sub-problem*, which performs content placement driven by the local demand.
- We introduce a heuristic that initially performs a  $K$ -means-inspired clustering of users on the content preference space, capturing also geographical constraints arising from the users' location. This serves as a *preparation phase* for the user association that will follow.
- For the user association sub-problem, our heuristic takes advantage of the resulting clusters and performs a content-preference aware user association, by solving an instance of Generalized Assignment Problem (GAP). In particular, the user clusters are utilized for the sophisticated *initialization* of GAP instance parameters. For the caching sub-problem, our heuristic performs a local demand-aware content placement by solving a 0-1 Knapsack Problems per SBS.
- We demonstrate the effectiveness of our heuristic through numerical results, evaluating it against schemes that follow association approaches that either lack a preparation phase or do not take into account the user geographical constraints. As it is evident, our clustering phase can increase the cache hit ratio up to 20%.

The rest of the paper is organized as follows. In section II we introduce our system model. In section III, we formulate the joint problem, through which the two sub-problem occurs, and in section IV we present our heuristic. In section V we demonstrate our evaluation results and in section VII we conclude the paper.

## II. SYSTEM MODEL

We assume that a set of SBSs  $\mathcal{S}$  and an MBS  $M$  work in conjunction, comprising a next generation mobile network,

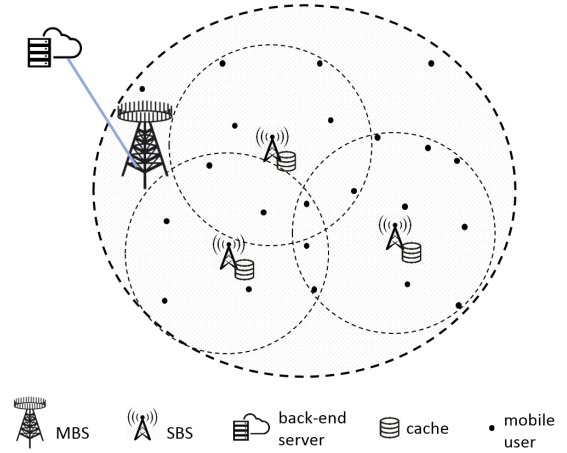


Fig. 1. An MBS and three SBSs work in conjunction to cover a certain geographic area, serving multiple users.

as depicted in Fig. 1. Each SBS can serve a set of mobile users within its range, while the MBS can serve any mobile user in the system. The system supports dual connectivity, hence a user is always connected to the MBS but he can simultaneously have a connection to one of the SBSs. A set  $\mathcal{U}$  of mobile users that access the Internet through the MNO's network, generate content item requests by browsing videos in the platform of a CP such as YouTube, Netflix or Amazon Prime Video. We assume that all SBSs are equipped with caches, thus are capable to store content items provisioned by the CP. Taking advantage of SBS's caches, content can be placed close to the users, thus leading to better user QoE and reduced backhaul link utilization and conservation of MBS's radio resources. The dimension of time in our system is captured by time-slots, i.e. time intervals of few hours. In each time-slot, we consider a "snapshot" of our system where users are assumed to be static, and their demands are assumed stationary stochastic processes.

**Users' association to SBSs.** Each mobile user  $u \in \mathcal{U}$  is by default associated to the MBS  $M$ , but he can as well be associated to at most one SBS  $s \in \mathcal{S}$  in his proximity. At a given time-slot, user  $u$  may be located within the range of multiple of SBSs. We use  $\mathcal{N}(u)$  to denote the SBSs that  $u$  can be associated with, i.e., the SBSs located in his neighborhood. When a user  $u$  is associated to an SBS  $s \in \mathcal{S}$ ,  $s$  serves as  $u$ 's *primary* source for fetching content, while the MBS is mostly responsible for the control plane. Note that users that can not be associated to any SBS will use MBS as their primary content source. Figure 1 shows a scenario where a geographic area is covered by an MBS and 3 SBSs. Some users fall in the range of more than one SBSs, thus certain association decisions should be taken.

**SBS power constraints.** Each SBS has a limited transmission power that is split among all users associated with it. The mobile users have QoS requirements that come from the CP platform, based on the "quality" of the acquired product (720p, 1080p etc.). This means that the MNO should guarantee

a minimum downlink data rate for each user. Hence, the SBS resources should be allocated in a way so that the QoS requirements of each associated user are satisfied. The cost of achieving the required downlink data rate is not the same for all users, as it is strongly related to the physical distance of the user from the associated SBS and link conditions. In particular, the achievable data rate  $r_{us}$  for a user  $u$  associated to an SBS  $s$  is an increasing and concave with respect to SINR (Signal-to-Interference-and-Noise-Ratio) function, which can be derived through Shannon's formula:

$$r_{us} = b_{us} \log_2(1 + \text{SINR}_{us}), \quad (1)$$

where  $b_{us}$  is the bandwidth that SBS  $s$  allocates to user  $u$ . Assuming that SBS  $s$  has a total bandwidth of  $B_s$ , the number of users that can be served is limited and depends on their SINR values for the given SBS. The SINR fades as the distance of a user for the SBS increases, or the noise and interference from other signal sources becomes higher. In this work, we assume that SINR for each user-SBS pair is given and the power of OFDMA sub-carriers is fixed. The MNO only controls the bandwidth that is allocated to each user for achieving the required data rate, *i.e.*, the number of sub-carriers. Thus, we assume that the bandwidth allocated to each user is an increasing function on his distance from the SBS. Finally, we assume that the only case where a user is associated only to the MBS is when all SBSs in his neighborhood have reached the  $B_s$  upper bound and cannot serve more devices.

**Content items.** We consider that a content catalogue  $\mathcal{I}$  is made available to users  $\mathcal{U}$ . Considering that there is a number of features  $F$  whose values denote the type of a content item or a mobile viewer, we assume that each item  $i \in \mathcal{I}$  is characterized by a vector  $\mathbf{p}_i \in [0, 1]^F$  of dimension  $F$ . The requested content items may have different sizes, ranging from large movie files to small advertisement clips, thus we use  $L_i, i \in \mathcal{I}$  to denote the size of item  $i$  in bytes.

**Cache-enabled SBSs.** We assume that each SBS  $s \in \mathcal{S}$  maintains a cache of storage capacity  $C_s$  bytes. Each cache stores a set of content items  $P_s$  (*cache placement*), which is a subset of the complete catalogue  $\mathcal{I}$ . The placement to each cache is determined once in each time-slot and can be only updated during the transition to the next time-slot. Recall that a time-slot can be of the order of few hours. The requested items that are not available in the local caches are served by the MBS  $M$  that fetches the content from a back-end remote server located to the cloud.

**Content demand.** Each user  $u \in \mathcal{U}$  is characterized by a feature vector  $\mathbf{p}_u \in [0, 1]^F$  of dimension  $F$ . Then, we can derive user's  $u$  preferences over all items  $i$  in the catalogue  $\mathcal{I}$ , by examining the cosine similarity of vector  $\mathbf{p}_u$  to each vector  $\mathbf{p}_i, \forall i \in \mathcal{I}$ . In other words, the following formula determines how "close" to the preferences of user  $u$  item  $i$  is:

$$\phi(\mathbf{p}_u, \mathbf{p}_i) = \frac{\sum_{f=1}^F \mathbf{p}_u(f) \mathbf{p}_i(f)}{\sqrt{\sum_{f=1}^F \mathbf{p}_u^2(f)} \sqrt{\sum_{f=1}^F \mathbf{p}_i^2(f)}} \quad (2)$$

By calculating the above for each user-item pair, we get a user-to-items similarity vector  $(\phi(\mathbf{p}_u, \mathbf{p}_1), \dots, \phi(\mathbf{p}_u, \mathbf{p}_{|\mathcal{I}|}))$  for each user  $u \in \mathcal{U}$ . This vector is normalized in order to extract user's  $u$  demand distribution  $d_u$  over all items in  $\mathcal{I}$ . We denote as  $d_u(i)$  the probability that item  $i \in \mathcal{I}$  is requested by user  $u$ , which is

$$d_u(i) = \frac{\phi(\mathbf{p}_u, \mathbf{p}_i)}{\sum_{j \in \mathcal{I}} \phi(\mathbf{p}_u, \mathbf{p}_j)}. \quad (3)$$

### III. PROBLEM FORMULATION

In this section we formulate the problem of jointly associating mobile users to SBSs and placing content in the SBS-located caches. Then, we decompose the joint problem into two sub-problems that are solved sequentially. First, a content preference-aware user association is determined. Then, a demand-driven content placement is decided for each SBS independently. As preparation step for solving the former, we propose a heuristic that takes advantage of *clustering* techniques to perform an initial grouping of mobile users with similar content preferences. The resulting user clusters are then utilized for achieving a similarity-driven association of users, based on their content preferences. Associating users with similar preference to the same SBS will boost the efficiency of content placement that will be performed by solving the later sub-problem, *i.e.* it will increase the overall cache hit ratio.

#### A. Joint User Association and Content Placement Problem

Our objective is to maximize the portion of the total users' demand satisfied by the content being cached at the SBSs. We use two sets of binary decision variables that determine users association and content placement,  $\mathbf{x} = \{x_{us}\}_{u \in \mathcal{U}, s \in \mathcal{S}}$  and  $\mathbf{y} = \{y_{is}\}_{i \in \mathcal{I}, s \in \mathcal{S}}$  respectively. If  $x_{us} = 1$ , user  $u$  is associated to SBS  $s$ , while  $x_{us} = 0$  otherwise. Respectively,  $y_{is} = 1$  means that item  $i$  is cached to SBS  $s$  collocated cache and  $y_{is} = 0$ , otherwise. Then, the joint content placement and user association problem is formulated as follows:

$$\max_{\mathbf{x}, \mathbf{y}} \sum_{u \in \mathcal{U}} \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}} x_{us} y_{is} d_u(i) \quad (4)$$

$$s.t. \quad \sum_{i \in \mathcal{I}} y_{is} L_i \leq C_s, \quad \forall s \in \mathcal{S} \quad (5)$$

$$\sum_{u \in \mathcal{U}} b_{us} x_{us} \leq B_s, \quad \forall s \in \mathcal{S} \quad (6)$$

$$\sum_{s \in \mathcal{N}(u) \cup M} x_{us} = 1, \quad \forall u \in \mathcal{U} \quad (7)$$

$$x_{us} \in \{0, 1\}, \quad u \in \mathcal{U}, \quad s \in \mathcal{S} \quad (8)$$

$$y_{is} \in \{0, 1\}, \quad i \in \mathcal{I}, \quad s \in \mathcal{S} \quad (9)$$

where constraints (5) and (6) reflect the limited cache capacity and bandwidth for each SBS, respectively. Constraint (7) captures the fact that a user can have only one primary content source, either one of the SBSs or the MBS. Note that the by-default contribution of MBS as secondary content source is not taken into account in the problem formulation. The above optimization problem has been shown to be NP-hard [8], hence we decompose into the two sub-problems that follow.

## B. Sub-Problems Formulation

In this section, we define the user association and caching sub-problems in such a way that their objectives are aligned with the objective of joint problem, i.e. they both aim at maximizing the portion of total demand satisfied by the SBS caches.

**User association sub-problem:** We decide the user association, so that the aggregate over all SBSs preference similarity of users associated to the same SBS is maximized. We cannot yet accurately quantify the cache hit ratio, as the cache placement will be determined in the next step. However, the association of users with similar demand characteristics to the same SBS will generate a more compact local demand, and indirectly boost the effectiveness of caching. We assume that each SBS  $s$  is characterized by a vector  $\mathbf{p}_s \in [0, 1]^F$ , which captures the ‘‘SBS content preferences’’. Note that vector  $\mathbf{p}_s$  is an input to our problem and can be initialized either by the preferences of all users being in range or by more sophisticated methods. In Remark 1, we elaborate more on the initialization options for vectors  $\mathbf{p}_s$  and our approach. The user association sub-problem is formulated as:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{u \in \mathcal{U}} \sum_{s \in \mathcal{S}} x_{us} \phi(\mathbf{p}_u, \mathbf{p}_s) \quad (10) \\ \text{s.t.} \quad & (6), (7), (8). \end{aligned}$$

The user-association sub-problem is an instance of the GAP, for which multiple known approximation algorithms exist.

**Caching sub-problem:** We decide the content placement so as to maximize the total demand satisfied by all caches, taking as input the solution of the association sub-problem. Using  $\mathcal{U}_s$  to denote the set of users associated to SBS  $s$ , the caching sub-problem is formulated as follows:

$$\begin{aligned} \max_{\mathbf{y}} \quad & \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \sum_{i \in \mathcal{I}} y_{is} d_u(i) \quad (11) \\ \text{s.t.} \quad & (5), (9). \end{aligned}$$

This formulation leads to solving multiple separate 0 – 1 Knapsack problems, one for each SBS.

**Remark 1.** While in the caching sub-problem we perform a content placement that is aware of users’ association, the association sub-problem cannot take into consideration the content placement that will follow. We alleviate this lack of awareness through a sophisticated initialization of each SBS preference vector  $\mathbf{p}_s$ . To do so, we introduce a *preparation phase* that initializes all SBS preference vectors through a method that is driven by user clustering. The heuristic we propose consists of two discrete phases:

- *Clustering phase.* It performs clustering of users based on the similarity of their content preferences, accounting for constraints raised by their location in the system.
- *Association and Caching phase.* First, it exploits the resulting user clustering to perform sophisticated users’ association by solving the *association sub-problem*. Then,

TABLE I  
NOTATION TABLE

Notation	Context
$\mathcal{I}, \mathcal{U}, \mathcal{S}$	Items, Users, SBSs
$M$	MBS
$F$	Number of features characterizing an item or a user
$\mathbf{p}_i$	Vector with item’s $i$ relevance to all thematic categories
$\mathbf{p}_u$	Vector with user’s $u$ preferences over all thematic categories
$\mathbf{p}_s$	Vector with SBS’s $s$ ‘‘preferences’’ over all thematic categories
$\phi(\cdot)$	Function that estimates cosine similarity between two vectors
$d_u$	Demand distribution of user $u$ over all items
$N(u)$	Set of SBSs that user $u$ can be associated to
$b_{us}$	Bandwidth of SBS $s$ allocated user $u$ (if $u$ is associated to $s$ )
$r_{us}$	Achievable downlink data rate for user $u$ if associated to SBS $s$
$B_s$	Total bandwidth of SBS $s$
$C_s$	Storage capacity of cache collocated with SBS $s$ in bytes
$\mathcal{P}_s$	Items placement in cache $s$
$L_i$	Size of item $i$ in bytes

it performs an association-aware content placement by solving the *caching sub-problem*.

## IV. OUR HEURISTIC

We introduce a pseudo-polynomial-time heuristic whose constituent procedures are: (i) a preparation step, where a location-aware clustering of users based on their content preferences takes place. (ii) the user association algorithm, that utilizes the resulting clusters of users for the sophisticated initialization and solution of the user-association sub-problem, and (iii) the content placement algorithm, that solves a 0 – 1 Knapsack problem per SBS.

### A. Clustering phase

We perform clustering of mobile users based on the similarity of their preferences over the different content thematic categories. User preferences are already available in CP platforms, thus minor effort is required for obtaining it. In particular, we generate user clusters following an approach inspired by the  $K$ -means clustering technique [9]. In our case, the geographical proximity of each pair of users should be taken into account together with the similarity of their preferences. We thus avoid clustering together users that cannot be served by the same SBS by virtue of proximity. This constraint can be extracted from the capabilities of SBSs. The main steps of the preparation phase are:

(i) **Location-constrained user similarity.** The similarity of each pair of users is evaluated by means of cosine similarity of their preference vectors. In particular, the similarity  $\phi(\mathbf{p}_u, \mathbf{p}_{u'}) \in [0, 1]$  between users  $u$  and  $u'$  can be extracted by (2). The calculation of cosine similarity for all user pairs, results to a matrix  $\Phi : |\mathcal{U}| \times |\mathcal{U}| \rightarrow [0, 1]$ . Considering that the geographic coverage of an SBS is determined by a radius  $\rho$ , we assume that it is not possible to associate two users in the same SBS if their distance is greater than  $2\rho$ . Given the coordinates  $(x, y)_u, (x, y)_{u'}$  for each pair of users  $u, u' \in \mathcal{U}$ , if the Euclidean distance of two user locations is greater than  $2\rho$ , we set element  $\Phi_{u, u'}$  to zero, i.e.,  $\sqrt{(x_u - x_{u'})^2 + (y_u - y_{u'})^2} > 2\rho \implies \Phi_{u, u'} = 0$ .

(ii) **Initial centroids.** In this step, we aim at creating a set  $\mathcal{K} = \{\kappa_1, \dots, \kappa_{|\mathcal{K}|}\}$  of clusters, where  $|\mathcal{K}| = |\mathcal{S}|$ . This modeling decision is discussed in Remark 2. To kick off the clustering process, we need to define  $|\mathcal{K}|$  initial centroids, by performing the following:

- We utilize the resulting matrix  $\Phi$  to calculate the *average preference similarity*  $\hat{\Phi}_u$  of each user to **all** other users in the system

$$\hat{\Phi}_u = \frac{\sum_{j \in \mathcal{U} \setminus \{u\}} \Phi_{j,u}}{|\mathcal{U}| - 1} \quad (12)$$

- We select the user with the highest average similarity  $u^* = \arg \max_u \hat{\Phi}_u$ , and we initialize the centroid of first cluster  $\kappa_1 \in \mathcal{K}$  with this user's content preferences, i.e.  $\mathbf{p}_{\kappa_1} = \mathbf{p}_{u^*}$ .

- For the remaining  $|\mathcal{K}| - 1$  cluster centroids, we follow an iterative process until all  $|\mathcal{K}|$  centroids have been determined. In each step of the process, one of the remaining users is selected to initialize a new centroid vector. The vector of the  $i$ -th cluster's centroid is determined by the following formula:

$$\mathbf{p}_{\kappa_i} = \left\{ \mathbf{p}_{u^*} \mid u^* = \arg \max_u \left[ \hat{\Phi}_u - \frac{1}{i-1} \sum_{j=1}^{i-1} \phi(\mathbf{p}_u, \mathbf{p}_{\kappa_j}) \right] \right\}. \quad (13)$$

The first term in the formula above, is the average similarity of user  $u$  with all others in the system, while the second term captures the average similarity of user  $u$  with the already initialized centroids of clusters  $\kappa_1, \dots, \kappa_{i-1}$ . The *intuition* behind formula (13) is that we would like to initialize the next cluster centroid with the preferences of a user that has a high average similarity with all other users, while its preferences significantly differ from the already initialized centroids.

(iii) **Clustering process.** With the initial centroids as input, we perform an iterative user clustering process with a rationale similar to that of  $K$ -means. Each iteration has two steps that are repeated until the users-to-clusters assignment converges, or until the maximum iteration is reached. *First*, we assign each user to his closest centroid, in terms of location-constrained preference similarity. The similarity of users to the different centroids is again evaluated by means of cosine similarity eq. (2). Thus, the ‘‘closest’’ centroid for user  $u$  is given by  $\kappa^* = \arg \max_{\kappa} \phi(\mathbf{p}_u, \mathbf{p}_{\kappa})$ , where  $\kappa \in \mathcal{K}$ . We use  $\mathcal{U}_{\kappa_i}$  to denote the set of users assigned to cluster  $\kappa_i$ . *Second*, we recalculate the mean preference vector for each of the resulting user clusters and we update the preference vector of respective centroids, including its geographic location in the system. In particular, the new centroid preference vector of cluster  $\kappa \in \mathcal{K}$  across all  $F$  thematic categories is  $\mathbf{p}_{\kappa} = (\mathbf{p}_{\kappa}(1), \dots, \mathbf{p}_{\kappa}(F))$  where  $\mathbf{p}_{\kappa_i}(f) = \frac{1}{|\mathcal{U}_{\kappa_i}|} \sum_{u \in \mathcal{U}_{\kappa_i}} \mathbf{p}_u(f)$ .

**Remark 2.** Recent literature [10] indicates that the *strict* assignment of entire cluster(s) to the SBS, combined with a low total number of clusters, may affect the efficiency of content placement. In that context, a system could determine cluster-SBS assignments that would lead to a slack loss in

---

### Algorithm 1: Clustering phase

---

**Input:** Preferences  $\mathbf{p}_u$  and coordinates  $(x, y)_u, \forall u \in \mathcal{U}$   
**Output:**  $|\mathcal{K}|$  user clusters, each cluster  $\kappa \in \mathcal{K}$  has users  $\mathcal{U}_{\kappa}$  and preference vector  $\mathbf{p}_{\kappa}$   
**Complexity:**  $\mathcal{O}(|\mathcal{U}|^2)$

---

*Location-constrained user similarity:*

```

1 foreach  $(u, u')$  do
2   if  $\sqrt{(x_u - x_{u'})^2 + (y_u - y_{u'})^2} < 2\rho$  then
3      $\Phi_{u,u'} = \phi(\mathbf{p}_u, \mathbf{p}_{u'})$ 
4   else
5      $\Phi_{u,u'} = 0$ 
6   end
7 end
8 Centroids' initialization:
9 foreach  $u$  do
10    $\hat{\Phi}_u = \frac{\sum_{j \in \mathcal{U} \setminus \{u\}} \Phi_{j,u}}{|\mathcal{U}| - 1}$ 
11 end
12  $\mathbf{p}_{\kappa_1} = \{\mathbf{p}_{u^*} \mid u^* = \arg \max_u \hat{\Phi}_u\}$ 
13 for  $i = 2 : |\mathcal{K}| - 1$  do
14    $\mathbf{p}_{\kappa_i} = \left\{ \mathbf{p}_{u^*} \mid u^* = \arg \max_u \left[ \hat{\Phi}_u - \frac{1}{i-1} \sum_{j=1}^{i-1} \phi(\mathbf{p}_u, \mathbf{p}_{\kappa_j}) \right] \right\}$ 
15 end

```

*Clustering Process:*

```

16 while  $\mathcal{U}_{\kappa}, \forall \kappa \in \mathcal{K}$  remain the same do
17   foreach  $u$  do
18      $\kappa^* = \arg \max_{\kappa} \phi(\mathbf{p}_u, \mathbf{p}_{\kappa}), \kappa \in \mathcal{K}$ 
19      $\mathcal{U}_{\kappa^*} = \mathcal{U}_{\kappa^*} \cup \{u\}$ 
20   end
21   foreach  $\kappa$  do
22     Update  $\mathbf{p}_{\kappa}$  with avg. preferences of users in  $\mathcal{U}_{\kappa}$ 
23   end
24 end

```

---

terms of bandwidths, i.e. the spare bandwidth of an SBS could remain unallocated since no unassigned user cluster would fit in this SBS. In our study, we follow an approach where the resulted cluster centroids (average content preferences) are only utilized to perform a sophisticated initialization of a GAP instance where the actual user association is performed. Especially, during this initialization, we set the ‘‘content preferences’’ of each SBS and we do not perform a strict association of entire clusters to SBSs. Eventually, users from multiple clusters may end-up to the same SBS. Consequently, our modeling decision to sets the number of cluster equal to that of SBSs ( $|\mathcal{K}| = |\mathcal{S}|$ ) will not lead into resources underutilization.

### B. Association and Caching phase

In this phase, we initially map the mean preference vectors of the resulting clusters to SBSs. However, this does **not** imply the association of this cluster's users to an SBS, but it is only

used for initializing  $\mathbf{p}_s$  for each SBS  $s \in \mathcal{S}$ . The main steps of the association and caching phase are:

(i) **Map cluster centroids' mean preference to SBSs.** We map the mean preference vector of each cluster centroid to an SBS taking into account both the benefit and the cost generated when the users of a cluster are associated to a certain SBS. We thus define a function  $V(\kappa, s)$  to estimate the value of mapping cluster  $\kappa$  to SBS  $s$ , as:

$$V(\kappa, s) = \sum_{u \in \mathcal{U}_\kappa} \phi(\mathbf{p}_u, \mathbf{p}_\kappa) - \hat{b}_{us}. \quad (14)$$

The first term denotes the *potential benefit* that will be generated if the users of the given cluster are associated to the given SBS. The second term denotes the respective *association cost* in terms of bandwidth. Note that  $\hat{b}_{us} \in (0, 1]$  is the normalized value of  $b_{us}$ . By applying formula (14) for each cluster-SBS pair, we generate a  $|\mathcal{K}| \times |\mathcal{S}|$  matrix  $\mathbf{V}$  that gives a rough estimate of the value of associating a cluster to each of the SBSs. Taking  $\mathbf{V}$  as input, we perform an one-to-one matching between the mean preference vector of each cluster and the SBSs by following the Hungarian method [11]. This results to the initialization of all vectors  $\mathbf{p}_s$ ,  $\forall s \in \mathcal{S}$ .

(ii) **Associate users to SBSs.** We associate users to SBSs by solving our *association sub-problem* (10), which is an instance of the GAP. In particular, users and SBSs are mapped to the items and agents/bins of the GAP, respectively. The *value* for associating a user  $u$  to an SBS  $s$  is given by estimating the similarity between the preference vectors of  $u$  and  $s$ , i.e.  $\phi(\mathbf{p}_u, \mathbf{p}_s) \in [0, 1]$ , while the respective *normalized association cost* is given by  $\hat{b}_{us} \in (0, 1]$ . To solve this GAP instance, we apply the Martello-Toth approximation scheme [12] that has a complexity of  $\mathcal{O}(|\mathcal{U}||\mathcal{K}| \log |\mathcal{K}| + |\mathcal{U}|^2)$ . This algorithm requires a *desirability* metric that drives the assignment. In most of the cases, the *desirability* matches the *value*. However it can also be a metric that combines both *value* and *cost*. In our heuristic, we define *desirability* as  $\phi(\mathbf{p}_u, \mathbf{p}_s) / \hat{b}_{us}$ .

(iii) **Local demand-aware caching.** Taking the resulting association as input, we determine the placement of content in each cache aiming to maximize the sum of local demand that will be served across caches. This results in the *caching sub-problem* which is tackled by solving a 0 – 1 Knapsack problem for each SBS by using any related algorithm, e.g., the pseudo-polynomial Dynamic Programming algorithm in [12]. The *value* of placing a content item  $i$  to the an SBS  $s$  is given by aggregating the demand probabilities over all the associated users to  $\mathcal{U}_s$ , i.e.  $\sum_{u \in \mathcal{U}_s} d_u(i) = \frac{\phi(\mathbf{p}_u, \mathbf{p}_i)}{\sum_{j \in \mathcal{I}} \phi(\mathbf{p}_u, \mathbf{p}_j)}$ . On the other hand, the *placement cost* of item  $i$  is determined by its size  $L_i$ .

**Remark 3.** The overall complexity of our heuristic is determined by the sum of the individual complexities of its three constituent procedures. The user clustering and the user association are two system-wide procedures that exhibit a polynomial time complexity of  $\mathcal{O}(|\mathcal{U}|^2)$  and  $\mathcal{O}(|\mathcal{U}||\mathcal{K}| \log |\mathcal{K}| + |\mathcal{U}|^2)$ , respectively. For the caching procedure, we solve multiple parallel 0 – 1 Knapsack problems, one per SBS-located

cache. Each caching procedure is performed by a Dynamic Programming algorithm, which implies a pseudo-polynomial time complexity  $\mathcal{O}(|\mathcal{I}|^2)$ , where  $|\mathcal{I}|$  is the size of item catalogue.

## V. EVALUATION

We demonstrate the performance of our heuristic against alternative schemes that either do not perform clustering or do not take into account the geographical constraint aspects. The schemes under comparison are: (i) Our heuristic. (ii) The JCA scheme, which is the algorithm proposed in [8], but without accounting for recommendations. This scheme follows a naive initialization of the GAP instance, that solves user association. In particular, it does not perform clustering of users as a preparation phase, instead the SBS preferences are built based on the preferences of all users in the proximity of an SBS. (iii) Our heuristic without taking into account the geographical constraints, neither during the clustering, nor during the association phases. (iv) A scheme that performs user association based on the proximity of users and SBSs. This scheme adopts the methods followed in association and caching phase of our heuristic, however the objective of the association problem is the minimization of association cost, hence it has the same complexity with our scheme. (v) A greedy algorithm that is briefly presented below.

**Greedy algorithm.** We first sort each user-item pair in decreasing order of valuation, thus creating a matrix of  $|\mathcal{U}| \cdot |\mathcal{I}|$  values. Then, we parse the matrix once and for each user-time pair we perform on of the following actions:

(a) If the user is associated, we check if the current item is stored in the cache of the respective SBS. If it is not, and there is enough cache capacity, we then store the item.

(b) If the user is not associated and there are available SBSs, we estimate the “score” for associating the user to each of the SBSs taking into account that a caching action may follow:

$$\begin{aligned} \text{Score} = & as_{pen} \cdot as_{val} - (1 - as_{pen}) \cdot as_{cost} + \\ & ca_{pen} \cdot ca_{val} - (1 - ca_{pen}) \cdot ca_{cost} \end{aligned}$$

The association value  $as_{val}$  captures the valuation of the current user for items that are already stored in the cache of the examined SBS. The caching value  $ca_{val}$  captures the valuation generated for the users that are already associated to the SBS if we store the current item. The association and cache penalties,  $as_{pen}$  and  $ca_{pen}$ , take into account the residual SBS data rate and cache capacity respectively. If the current item is already cached, the second part of the formula is omitted. The user with the higher score is associated.

**Simulation Parameters.** We set up multiple instances of our system by setting different values on the key parameters. The evaluated instances consist of an MBS and multiple SBSs (4 to 50) serving a certain geographic region (10000 meters<sup>2</sup>), where mobile users (50 to 500) are randomly scattered. Users are requesting content items of variable size (4 to 6 MB) from a catalogue (20 to 1000 items) based on their preferences. Both mobile users and content items are characterized by a feature vector of size 8 that follows a Zipf distribution. Each



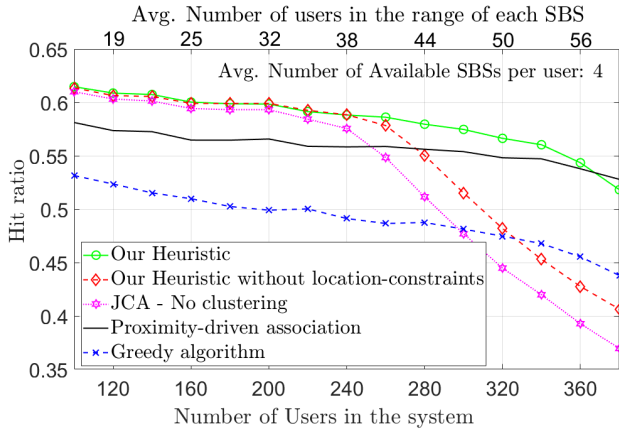


Fig. 2. Impact of total number of users in the system. Hit ration of all schemes in a set up of: 1 MBS, 25 SBSs each of them able to serve from 7 up to 35 users, 100 total items, SBS density of 4 SBSs per user, cache size that fits 40% of total items and number of total users ranging between 100 and 400.

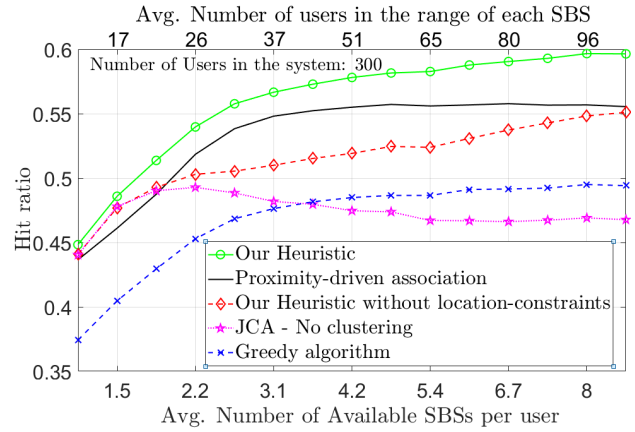


Fig. 3. Impact of SBS density, i.e. average number of available SBSs per user. Hit ratio of all schemes in a set up of: 1 MBS, 25 SBSs each of them able to serve from 7 up to 35 users, 300 total users, 100 total items, cache size that fits 40% of total items and SBS density from 1 up to 8.5.

SBS can support a certain number of users (5 to 75) in his range (10 to 85 meters) due to bandwidth limitations. The SBS caches can store a limited number of items (10% up to 85% of catalogue). By setting different values on these parameters, we can evaluate the performance of all schemes under different levels of network/users' density and for system dimensioning.

**Evaluation Metric.** We use the percentage of the global demand to be satisfied by the SBS-collated caches as the evaluation metric for all schemes. We also use the term “*hit ratio*” to refer to this metric. Note that users being associated only to the MBS are also considered in the calculation of this metric. The *hit ratio* for these users is 0. It is worth mentioning that even a more % of improvement in overall cache hit ratio, may result to significant cost reduction for the MNO.

**Evaluation Results.** The numerical results reveal that our heuristic achieves superior performance (increase cache hit ratio up to 20%) compared to all other schemes in most of the cases. In particular:

- (i) *Number of users.* For instances with only few users in the system, our heuristic achieves a *hit ratio* which is very close to the ones achieved by the *JCA-No clustering* and *Our heuristic without location-constraints* schemes. However, as shown in Fig. 2, when the total number of users in the system increases, the performance of these two schemes significantly decreases. We can also observe that as the total number of users increases and the system is stressed, i.e. the SBSs cannot serve a percentage higher than 4% of users in the system, the *proximity-driven* scheme has the best performance since it “fits” more users to SBSs.
- (ii) *SBSs' density.* Figure 3 demonstrates that as the number of association options per user increases, our heuristic achieves even higher hit ratio improvement compared to the *proximity-driven* scheme, while it still significantly outperforms all other schemes.
- (iii) *Users' density.* As shown both in Fig. 2 and Figure 3, when the average number of users located in the range

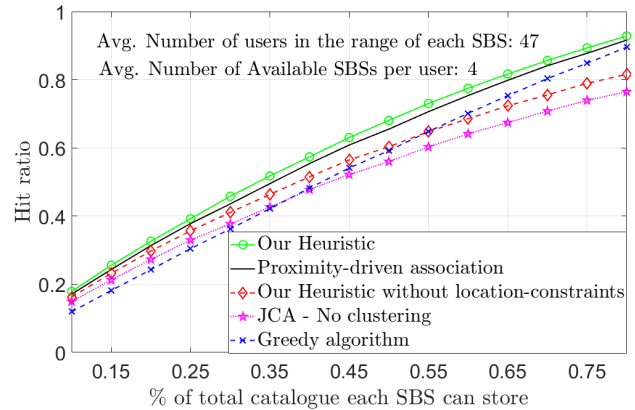


Fig. 4. Impact of cache size. Hit ratio of all schemes in a set up of: 1 MBS, 25 SBSs each of them able to serve from 7 up to 35 users, 100 total items, 300 total users, SBS density of 4 SBSs per user and SBS cache size that can store from 10% up to 80% of the total item catalogue.

of a single SBSs increases, the *JCA-No clustering* scheme faces performance deterioration due to its naive approach on initializing the SBS feature vector, i.e.  $\mathbf{p}_s$ .

(iv) *Cache size.* Figure 4 shows that as the cache size increases, our heuristic maintains better performance than all other schemes. Interestingly, the performance of the *greedy algorithm* becomes higher than the *CA-No clustering* and *Our heuristic without location-constraints* schemes when the cache size approaches a value that is higher than 40% and 55% of total catalogue, respectively.

## VI. RELATED WORK

There are several studies for joint content placement and user association [4], [8], [13]–[18]. Their objectives focus on the minimization of the content access delay, minimization of the aggregate operational cost to serve all the incoming request or maximization of cache hit ratio. The authors in [13]–[15] attempt to solve the join caching and association problem by applying the McCormick envelopes and the Lagrangian partial

relaxation methods. An issue with these solutions is the large number of repetitions that need to be made in order to ensure convergence. The authors in [16] attempt to solve the problem with an iterative two step algorithm. In the first step, the association is considered as fixed and they optimally determine the placement of content. In the second step, the optimal mobile user association is determined over fixed caches. A similar approach is followed in [8], and a third step with content items' recommendation to users is also considered. However, both [8], [16] lack extensive simulation results. The authors in [17] perform a demand-aware user association to SBSs, i.e. each SBS decide which user to serve based on the local content availability and the data rates it can delivers.

The authors in [4], [18] consider clustering methods to achieve efficient association of mobile users to the SBSs. Specifically, in [4], a two-phase solution is proposed. First, the authors consider a clustering algorithm that groups users with similar content demand in order to associate them in the same SBS. Then, a reinforcement learning algorithm is proposed in order to learn the popularity distribution of contents requested by its group of users to optimize its caching strategy. The authors in [18] use a  $k$ -NN approach to cluster users with similar demands and entirely associates different groups of users to different SBSs. These studies do not use geographical constraints during clustering process and do not dig into details on how matching of clusters to SBSs is performed. Methods for estimating the content items' similarity has been proposed for similarity caching [19]. However, in our problem we aim at serving the users' actual demand and do consider offering of alternative items when the requested one is not available. Finally, a comparative analysis of multiple initialization methods for the  $K$ -means algorithm has been presented in [20].

In this work, we decompose the joint problem into two sub-problems that are solved sequentially, where the first one focuses on the content preference-aware user association while the second one handles the demand-aware caching per SBS. Contrary to [4], [18], during the clustering process, our work takes into account geographical limitations that may arise from the users' location and defines a method for mapping user clusters to SBSs. Also, as shown in section V, our heuristic outperforms algorithms that follows naive approaches for capturing user preferences during the association process.

## VII. CONCLUSIONS

In this work, we decomposed the NP-hard problem of joint control over user association and content placement into to simpler sub-problems. We proposed a heuristic that solves the two sub-problems sequentially taking advantage of clustering techniques and approximation algorithms for the well-know GAP and Knapsack problem. The results revealed that our heuristic outperformed schemes that are agnostic to the preferences of users and to geographical constraints. As future steps, we would like to evaluate the performance of our heuristic at a larger scale and under real system conditions, and to study the problem in a setup that captures the mobility of users and cost of cache updates.

## VIII. ACKNOWLEDGMENT

This research has been funded by the Operational Program “Human Resources Development, Education and Lifelong Learning”, co-financed by European Union (EU) and Greek national funds through project VELOS. I. Koutsopoulos acknowledges the support from a GSRT Research Reinforcement grant for the EU R&D project netCommons, and from the AUEB grant “Original Scientific Publications 2019”.

## REFERENCES

- [1] C. V. N. Index, “Global mobile data traffic forecast update, 2017–2022 white paper.” *Cisco: San Jose, CA, USA*, 2019.
- [2] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed caching helpers,” *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec 2013.
- [3] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5g wireless networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug 2014.
- [4] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, “Content-aware user clustering and caching in wireless small cell networks,” in *International Symposium on Wireless Communications Systems (ISWCS)*, 2014, pp. 945–949.
- [5] L. E. Chatzileftheriou, M. Karaliopoulos, and I. Koutsopoulos, “Caching-aware recommendations: Nudging user preferences towards better caching performance,” in *Proc. of IEEE INFOCOM 2017*, Atlanta, USA, May 2017, pp. 784–792.
- [6] P. Blasco and D. Gündüz, “Learning-based optimization of cache content in a small cell base station,” in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 1897–1903.
- [7] Y. Zhang, “User mobility from the view of cellular data networks,” in *Proc. of IEEE INFOCOM 2014*, 2014.
- [8] L. E. Chatzileftheriou, G. Darzanos, M. Karaliopoulos, and I. Koutsopoulos, “Joint user association, content caching, and recommendations in wireless edge networks,” in *In proc. of IFIP Performance*, 2018.
- [9] S. Lloyd, “Least squares quantization in pcm,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [10] L. Gkatzikis, V. Sourlas, C. Fischione, I. Koutsopoulos, and G. Dán, “Clustered content replication for hierarchical content delivery networks,” in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5872–5877.
- [11] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [12] S. Martello, “Knapsack problems: algorithms and computer implementations,” *Wiley-Interscience series in discrete mathematics and optimization*, 1990.
- [13] Q. Chen, H. Chen, R. Chai, and D. Zhao, “Network utility optimization-based joint user association and content placement in heterogeneous networks,” *EURASIP Journal on Wireless Communications and Networking*, 2018.
- [14] Y. Wang, X. Tao, X. Zhang, and G. Mao, “Joint caching placement and user association for minimizing user download delay,” *IEEE Access*, vol. 4, pp. 8625–8633, 2016.
- [15] R. Chai, Y. Li, Q. Chen, and C. Jin, “Joint user association and content placement in d2d-enabled heterogeneous cellular networks,” in *PIMRC*, 2018.
- [16] B. Dai and W. Yu, “Joint user association and content placement for cache-enabled wireless access networks,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing 2016, Shanghai, China, March 20-25, 2016*, pp. 3521–3525.
- [17] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, “Cache-aware user association in backhaul-constrained small cell networks,” in *IEEE WiOpt 2014*, 2014.
- [18] A. V. Ribeiro, L. N. Sampaio, and A. Ziviani, “Affinity-based user clustering for efficient edge caching in content-centric cellular networks,” in *ISCC*, 2018.
- [19] M. Garetto, E. Leonardi, and G. Neglia, “Similarity caching: Theory and algorithms,” in *Proc. of the IEEE INFOCOM 2020*, 2020.
- [20] M. E. Celebi, H. A. Kingravi, and P. A. Vela, “A comparative study of efficient initialization methods for the k-means clustering algorithm,” *Expert systems with applications*, vol. 40, no. 1, pp. 200–210, 2013.