Practical scheme for MISO Cache-Aided Communication

Itsik Bergel
Faculty of Engineering
Bar-Ilan University
Ramat Gan, Israel
Email: itsik.bergel@biu.ac.il

Soheil Mohajer
Department of ECE
University of Minnesota
Minneapolis, USA
Email: soheil@umn.edu

***Abstract*—Multi-antenna cache aided communication schemes suffer from high implementation complexity, due to the need of dividing each file into many small segments (known as the sub-packetization problem). In a recent publication, we introduced a low complexity scheme which significantly reduces the complexity, but works only for specific combinations of system parameters, including memory size, number of users, and number of antennas at the base station. In this work we present a novel extension that achieves the same number of degrees of freedom (DoF) at low complexity for any system parameter. In this scheme, a base station with multiple antennas serves multiple single-antenna mobiles, while using a small cache memory at each mobile to boost the network throughput. We also present a greedy scheduling algorithm, and show that it reaches nearly 100% of efficiency for a large number of users. The significant reduction in complexity and the general applicability make this scheme suitable for practical implementation.**

***Index Terms*—cache-aided communication, MISO, sub-packetization level, complexity**

I. Introduction

The popularity of large data and high speed applications continuously increases and demands more and more bandwidth. A unique characteristic of these applications is their variation over time, which results in a heavy network traffic in peak-hours, compared to the average network traffic.

Recent caching techniques introduced in [1] allow for two separate gains: (1) a *local gain* (or traditional caching gain) due to the fraction of the desired data of each user which is cached by that user and does not need to be transmitted, and (2) a *global gain* due to the interference cancellation and broadcasting opportunities provided by availability of the desired data of one user at other users' cache. While the local gain is typically small, the global gain scales with the aggregate size of the cache distributed among all the users in the network, and can be substantial due the availability of many devices, each with small memory.

The so called *coded caching* or *cache-aided communication* consists of a placement phase and a delivery phase. [1]. During the placement phase, prior to knowing the users' requests, we can pre-fetch and store at each users' memory some packets from the files in the database. During the delivery phase, after the requests are revealed, the server generates a set of coded messages and transmits them to all the receivers. All users should be able to decode their desired file from the received signal and their cache contents. The key feature of coded caching is the fact that caching a packet at one user provides an opportunity for *multicasting combined packets*, even if it is only requested by another user. This leads to a an achievable degrees of freedom (DoF), proportional to the number of cached copies of each piece of data among all the users.

In a network where the transmitter is equipped with L transmit antennas, it can simultaneously serve a number of users, and L degrees of freedom can be achieved. Interestingly, we can simultaneously benefit from the spatial diversity gain and caching gains. More precisely, it was shown by Shariatpanahi *et al.* that $L+M$ degrees of freedom can be achieved in a broadcast system with L transmit antennas at the server and an aggregate cache size that can distributedly store M complete copies of the database across the users [2]. Yet, this approach is not practical due to the need to divide each file to a large number of segments (termed sub-packetization level). More precisely, the scheme of Shariatpanahi *et al.* requires dividing each file into $\binom{U}{M}\binom{U-M-1}{L-1} = O\left(U^{M+L-1}\right)$ file segments, where U is the number of mobile users in the network [2]. This is a practically infeasible number, specially since cache aided communication is attractive mostly for networks with large number of users.

The problem of sub-packetization in cache-aided communication was mostly studied for the single antenna setting [3]–[6]. For a MISO system, Lampiris and Elia [7] proposed a placement and delivery scheme based on grouping and cache replication ideas, which reduces the sub-packetization as if the *effective* number of users is U/L. However, the drawback of this scheme is the requirement of the number of copies of the database distributedly cached across the users being divisible by the number of antennas, i.e., $L|M$. In practice, M is typically small, and hence such a constraint is not feasible.

In a recent work [8], we proposed a scheme that employs uncoded transmission, and achieves a sub-packetization level that is identical to that of a single antenna ($L=1$). Yet, the use of uncoded transmission requires much more transmission power. While this scheme achieves the same DoF, it requires

The work of S. Mohajer is supported by the National Science Foundation under grant CCF-1749981.

978-1-7281-5478-7/20/$31.00 ©2020 IEEE

a power that is $M+1$ times higher than that of Shariatpanahi *et al.* scheme in [2].

An alternative mthod is presented in [9], that combines the benefits of both schemes, by sending coded packets. This scheme achieves the sub-packetization level of a single antenna, with a negligible power loss (due to a lower diversity order). However, this scheme was so far derived only for the specific set of parameters, where $M+1$ divides $M+L$. This is a significant drawback, as it precludes the use of this scheme in many multi-antenna systems.

In this work, we present an extension of the scheme in [9] that can be applied to any number of antennas. This scheme suffers a negligible power reduction, but maintains the sub-packetization order of a single antenna system. That is, our proposed scheme requires a sub-packetization level of only $\binom{U}{M} = O\left(U^M\right)$, which is significantly smaller than $O\left(U^{M+L-1}\right)$ required by the scheme of [2]. Moreover, unlike the scheme of [2] where users should simultaneously decode multiple messages in a multiple access channel (MAC), in the proposed scheme each user receives and decodes only one message at each time.

Notation. We use bold letters (e.g. $\mathbf{h}$) to denote vectors and calligraphic letters to denotes sets. For two sets $\mathcal{A}$ and $\mathcal{B}$, we denote by $\mathcal{B}\backslash\mathcal{A}$ the set of all elements of $\mathcal{B}$ that do not appear in $\mathcal{A}$. We also use $\mathcal{A}\cup\mathcal{B}$ and $\mathcal{A}\cap\mathcal{B}$ to denote the union and intersection of $\mathcal{A}$ and $\mathcal{B}$, respectively. $|\mathcal{A}|$ denotes the size of set $\mathcal{A}$, and $\varnothing$ indicates an empty set. For two integers n and k we have $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Moreover, $[n]$ denotes the set $\{1,2,\ldots,n\}$. For a real number α, we denote its floor and ceiling by $\lfloor\alpha\rfloor$ and $\lceil\alpha\rceil$, respectively. The average is denoted by $\mathbb{E}[\cdot]$, and all the logarithms are in base 2.

II. System Model

We consider a network with U users, each with one receive antenna, and a single base-station (BS) which is equipped with L transmit antennas. We focus on a wideband communication scenario, in which the bandwidth, B, is divided into K frequency bins (e.g., OFDM), and each bin $k\in[K]$ carries one modulated symbol at a time, without inter-symbol interference. The received sample after matched filtering for the k-th frequency bin at user u is given by

$$y_{k,u}(t) = \mathbf{h}_{k,u}\mathbf{x}_k(t) + z_{k,u}(t), \tag{1}$$

where $\mathbf{x}_k(t)\in\mathbb{C}^{L\times 1}$ is the transmit vector at time slot t, $z_{k,u}\sim\mathbb{CN}\left(0,\frac{B}{K}\right)$ is the additive white Gaussian noise at User u in the frequency bin k, and $\mathbf{h}_{k,u}\in\mathbb{C}^{1\times L}$ is the channel vector from the BS to user u in the frequency bin k. We assume the BS has a total power constraint of P, and perfect channel state information is available at the BS. Each link between two antennas experiences fading, and we consider a *homogeneous* setting, where the channel of all users are statistically identical. More precisely, we assume $\mathbf{h}_{k,u}\sim\mathbb{CN}(\mathbf{0},\mathbf{I})$.

The BS has a dictionary of N files, namely $\{W_1,W_2,\ldots,W_N\}$, each of size F bits. Each user is interested in one of the files, chosen at random. Each user, u, is equipped with a memory Z_u that can store up to MNF/U bits. That is, M copies of the entire dictionary of the files can be distributedly stored across the users.

Cache *placement*, the process of filling the storage of the users with partial information from the dictionary, takes place before the users' demands are revealed. Later, each user u requests one file, say W_{d_u}, from the dictionary, and then the BS serves the users during the *delivery phase*. At the end of the delivery phase each user u should be able to decode W_{d_u} using the signal received from the BS and its cache Z_u.

III. The Proposed Scheme

The placement strategy in this work is similar to that of [1]. For an integer M, we first split each file into $\binom{U}{M}$ segments, and label them with subsets of $[U]$ of size M. Thus, the segments of file W_i are denoted as $W_{i,\mathcal{S}}$, where $\mathcal{S}\subseteq[U]$ and $|\mathcal{S}|=M$. During the off-peak time of the network (e.g., overnight), the network transmits data to fill the cache of all users. The cache of user $u\in[U]$ will be filled by all file segments whose label contain u, that is,

$$Z_u = \{W_{i,\mathcal{S}} : u\in\mathcal{S}, i\in[N]\}. \tag{2}$$

The transmission phase starts after the users send their file requests. The transmission phase is composed of multiple time slots. In each time slot the BS serves $M+L$ users simultaneously, such that each active user can collect some new information about its requested file. We define $\alpha = \frac{M+L}{M+1}$, and consider two separate cases in the following.

Case I (Integer α): The transmission is based on serving α groups in each time slot, where each group consists of $M+1$ users. We first define $\mathcal{G}$ to be the set of all subsets of $[U]$ of size $M+1$, that is, $\mathcal{G}=\{\mathcal{B}\subseteq[U], |\mathcal{B}|=M+1\}$. For each time slot, we choose α disjoint groups, and for a given group $\mathcal{B}$ we form a coded message

$$W_{(\mathcal{B})} = \bigoplus_{v\in\mathcal{B}} W_{d_v,\mathcal{B}\backslash\{v\}},$$

and send it along the direction that is orthogonal to the $(\alpha-1)(M+1)=L-1$ users in the other groups that are served in this time slot. Using L antennas in the BS, this direction is unique. Hence, this message will be received only at users in the group $\mathcal{B}$. Each user $u\in\mathcal{B}$ has all the file segments in $W_{(\mathcal{B})}$ except $W_{d_u,\mathcal{B}\backslash\{u\}}$, and hence can decode a segment of its desired file. Thus, the interference due to the users in the group is removed by user's cache and that due to users in other groups is canceled by zero-forcing (ZF).

The scheme proposed above requires an allocation of $\binom{U}{M+1}$ groups in $\mathcal{G}$ into collections of α *non-overlapping* groups, to be simultaneously served in each time slot. Such an allocation is presented in [10] for the case of $M=1$, but is not known in general. We present an efficient greedy algorithm in the next section, that solves the allocation problem for general α.

Case II (Non-integer α): When α is not integer, the proposed scheme still serves $M+L$ users simultaneously, except that the allocation becomes more complicated. In this case, the BS serves the users in $\lfloor\alpha\rfloor$ groups as before. However, in order to maintain the optimum degrees of freedom, it serves

an additional $\beta = M+L-(M+1)\lfloor\alpha\rfloor$ users. These additional users will be chosen from one or two *incomplete* groups. In the following we present an allocation scheme that is general and applicable to any value of α.

The delivery phase consists of n_{T} time blocks. The scheduling for block $n \in [n_{\mathrm{T}}]$ is determined by its allocation $\mathcal{T}[n]$. Each allocation $\mathcal{T}[n]$ is a collection of group allocations, where a group allocation is a pair $(\mathcal{A}, \mathcal{B})$. In such a pair, $\mathcal{B} \in \mathcal{G}$ is subset of $[U]$, and $\mathcal{A} \subseteq \mathcal{B}$ represents the users in the group that are actually served. Each user $u \in \mathcal{A}$ will decode the file segment $W_{d_u,\mathcal{B}\setminus\{u\}}$. Thus, if the whole group is served we have $\mathcal{A} = \mathcal{B}$, and $\mathcal{A} \subset \mathcal{B}$ represents the transmission to an incomplete group. We also denote by $\mathcal{U}[n]$ the set of all users that are served time slot n, which is given by

$$\mathcal{U}[n] = \bigcup_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[n]} \mathcal{A}. \tag{3}$$

For a pair $(\mathcal{A}, \mathcal{B}) \in \mathcal{T}[n]$, we define its coded message

$$W_{(\mathcal{A},\mathcal{B})} = \bigoplus_{u\in\mathcal{A}} W_{d_u,\mathcal{B}\setminus\{u\}}. \tag{4}$$

Then, this coded message is modulated to a codeword $\mathcal{W}_{(\mathcal{A},\mathcal{B})}$ of length τK. We further split this codeword into K chunks, each of length τ, and the chunk $\mathcal{W}_{k,(\mathcal{A},\mathcal{B})}$ will be transmitted over the frequency bin k, for $k \in [K]$. The transmit signal at time slot n and frequency bin k is formed by

$$\mathbf{X}_k[n] = \sum_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[n]} \sqrt{\frac{P}{\alpha K}} \mathbf{h}^{\perp}_{k,\mathcal{U}[n]\setminus\mathcal{B}} \mathcal{W}_{k,(\mathcal{A},\mathcal{B})}, \tag{5}$$

where $\mathbf{X}_k[n] = \begin{bmatrix}\mathbf{x}_k((n-1)\tau+1) & \cdots & \mathbf{x}_k(n\tau)\end{bmatrix}$ denotes the sequence of transmit signals $\mathbf{x}_k(t)$'s which are sent during the n-th time block at frequency bin k. Here, $\mathbf{h}^{\perp}_{k,\mathcal{D}} \in \mathbb{C}^{L\times 1}$ is a unit length beamforming vector which is orthogonal to the channel vectors of all users in $\mathcal{D}$ in frequency bin k, that is,

$$\mathbf{h}_{k,u}\mathbf{h}^{\perp}_{k,\mathcal{D}} = \mathbf{0}, \qquad \forall u \in \mathcal{D}, 1 \le k \le K. \tag{6}$$

Note that this is feasible only if $|\mathcal{D}| \le L-1$, since the BS has L antennas.

An allocation is called *valid* if it allows each active user to decode its requested file. Using the definition of allocation given above, the following theorem provides the conditions for a valid allocation.

Theorem 1: An allocation $\mathcal{T}[n]$ for $1 \le n \le n_{\mathrm{T}}$ is valid if all the following hold:

(1) For any $\mathcal{B} \subset [U]$ with $|\mathcal{B}| = M+1$ we have

$$\mathcal{B} = \bigcup_{n=1}^{n_{\mathrm{T}}} \bigcup_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[n]} \mathcal{A}. \tag{7}$$

(2) For the set of all active users we have

$$|\mathcal{U}[n]| = \sum_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[n]} |\mathcal{A}|. \tag{8}$$

(3) For any $(\mathcal{A}, \mathcal{B}) \in \mathcal{T}[n]$, we have

$$|\mathcal{U}[n] \setminus \mathcal{B}| \le L-1. \tag{9}$$

Proof: To prove the validity of an allocation we need to verify that each user u can decode all the segments of its requested file, W_{d_u}. Recall that file segments are labeled as $W_{d_u,\mathcal{S}}$, and note that $W_{d_u,\mathcal{S}}$ is cached at user u whenever $u \in \mathcal{S}$. Hence, for user u all segments of the form $W_{d_u,\mathcal{S}}$ with $u \notin \mathcal{S}$ should be transmitted and decoded. Note that for a required file segment $W_{d_u,\mathcal{S}}$, the set $\mathcal{S}\cup\{u\}$ is a group in $\mathcal{G}$. Therefore, the transmission condition is guaranteed by going over all groups $\mathcal{B} \in \mathcal{G}$ of size $M+1$ and making sure that each member v of each group $\mathcal{B}$ is served by the file segment $W_{d_v,\mathcal{B}\setminus\{v\}}$. Note that, whenever a pair $(\mathcal{A}, \mathcal{B})$ appears in an allocation, all users in $\mathcal{A}$ receive their corresponding file segment, and hence, the latter requirement is equivalent to Condition (1).

To ensure that the user can decode its desired segment, we need to verify two conditions: (i) that each user needs to decode at most one segment in each time slot, and (ii) such a segment is received with no interference. Condition (2) guarantees the former, by making sure that $\mathcal{A}$'s in each time block are non-overlapping and hence, each user will appear at most once as an *active* user in each time slot.

Finally, a message $W_{(\mathcal{A},\mathcal{B})}$ will be sent if requested by at least one user $u \in \mathcal{A} \subseteq \mathcal{B}$. This message is cached at all users in $\mathcal{B}\setminus\{u\}$. The interference at the remaining users will be removed by zero forcing. Condition (3) implies that this ZF is feasible using the L transmit antennas. ∎

Using Theorem 1, we observe that the maximum number of users served at one time block is $M+L$, while we have to serve all groups $\mathcal{B}$ of size $M+1$. Thus, the minimum required number of time blocks is

$$n_{\mathrm{T}} \ge \left\lceil \frac{\binom{U}{M+1}(M+1)}{M+L} \right\rceil. \tag{10}$$

The achieved number of DoF is given by

$$\mathrm{DoF} = \frac{U\binom{U}{M}(1-\frac{M}{U})}{n_{\mathrm{T}}} \le M+L. \tag{11}$$

Considering the lower bound in (10), equality in (11) can only be achieved if α divides $\binom{U}{M+1}$. Yet, we observe that the inequality will be tight for any value of M and L if the number of users, U, is large enough.

Example. Consider a system with $U = 6$ users, $L = 2$ antennas and $M = 2$ copies of the database cached across the users. A group allocation for this system is given by:

$$\begin{aligned}
\mathcal{T}[1] &= \{(\{4\},\{456\}),(\{356\},\{356\})\}\\
\mathcal{T}[2] &= \{(\{5\},\{456\}),(\{246\},\{246\})\}\\
\mathcal{T}[3] &= \{(\{6\},\{456\}),(\{145\},\{145\})\}\\
\mathcal{T}[4] &= \{(\{1\},\{123\}),(\{236\},\{236\})\}\\
\mathcal{T}[5] &= \{(\{2\},\{123\}),(\{135\},\{135\})\}\\
\mathcal{T}[6] &= \{(\{3\},\{123\}),(\{124\},\{124\})\}\\
\mathcal{T}[7] &= \{(\{3\},\{346\}),(\{146\},\{146\})\}\\
\mathcal{T}[8] &= \{(\{4\},\{346\}),(\{136\},\{136\})\}\\
\mathcal{T}[9] &= \{(\{6\},\{346\}),(\{345\},\{345\})\}\\
\mathcal{T}[10] &= \{(\{1\},\{125\}),(\{256\},\{256\})\}
\end{aligned}$$

$$\mathcal{T}[11] = \{(\{2\},\{125\}),(\{156\},\{156\})\}$$
$$\mathcal{T}[12] = \{(\{5\},\{125\}),(\{126\},\{126\})\}$$
$$\mathcal{T}[13] = \{(\{2\},\{234\}),(\{134\},\{134\})\}$$
$$\mathcal{T}[14] = \{(\{3\},\{234\}),(\{245\},\{245\})\}$$
$$\mathcal{T}[15] = \{(\{4\},\{234\}),(\{235\},\{235\})\}. \quad (12)$$

Note that in time slot $n = 1$ we serve four users, including a complete group $\{3, 5, 6\}$ and user 4 from an incomplete group $\{4, 5, 6\}$. The remaining users in this group will be served in time blocks $n = 2$ and $n = 3$. This allocation achieves the bound in (10). □

Maximizing the DoF of the system is equivalent to finding an allocation whose length, n_{T}, is close to the lower bound in (10). This task is not trivial, and an optimal solution is known only for integer α and $M = 1$ [10]. In the following section, we present a greedy algorithm which is shown to perform very close to the upper bound in (11) for a large number of users.

IV. Greedy Allocation Algorithm

In the following, we present a greedy approach for group allocation that tries to squeeze as many groups as possible into each time block. In each time block we serve at most $\lfloor\alpha\rfloor$ complete groups together with some additional users from one or two other groups if $\beta > 0$. We refer to the groups that are partially served in a given time block as *incomplete groups*.

The algorithm first chooses the incomplete groups, and then tries to fill the rest of the allocation. The variable *SplitRemaining* represents the users in the incomplete group that are not served in the current time block, and still need to be served. In the next time block, we choose β users from this set. Whenever this set is empty, (`SECTION 2`) the algorithm chooses a new group from $\mathcal{G}$ as an incomplete group to split, and serve over mutiple blocks.

For each new time block the algorithm first allocates some of the users in the selected *SplitGrp* (`SECTION 3`). Then it tries to add complete groups (`SECTION 4`) that obey the conditions of Theorem 1 (that is, there must be no overlap between the active users, and the first complete group must contain all non-active users of the split group).

If there are no more groups that can fit into the current time block, then the allocation for the time block is done, and the algorithm moves to a new block (`SECTION 5`). However, if there is no match for even a single complete group, then it is better to transmit all remaining split group at once.

At the last step before starting a new time block, there may be a possibility to squeeze more users (`SECTION 1`). If the time slot already occupies part of a split group and $\lfloor\alpha\rfloor$ complete groups, and the number of active users is still below $M+L$, then we try to add a part of an additional group (again according to Theorem 1). This can happen only in the time blocks where serving a split group is completed.

The selection of the best group in each stage is done greedily, based on an intuitive score that chooses the group whose elements have the minimum number of appearance up to the current stage of the algorithm (`SECTION 6`). This is implemented in function `ChGrp` given in Alg. 2.

Input: M, L, U
Output: $\mathcal{T}[n], n_{\mathrm{T}}$
Initializiation:
  $\tilde{\mathcal{G}} = \{\mathcal{B} \subseteq [U] : |\mathcal{B}| = M + 1\}$
  SplitRemaining $= \varnothing$
  SplitGrp $= \varnothing$
  NewSlot $=$ `True`
  $n_{\mathrm{T}} = 0$
  $\mathcal{T}[\,] =$ Empty array of allocations
  Residual $= 1 + ((M + L - 1) \mod (M + 1))$
while $|\tilde{\mathcal{G}}| > 0$ **do**
  if *NewSlot* **then**
    NewSlot $=$ `False`
    if $n_{\mathrm{T}} > 0$ and $0 < M + L - |\mathcal{A}| < M + 1$ **then**
    `%SECTION 1`
      NeededOverlap $= M + L - |\mathcal{A}|$
      SplitGrp $=$ `ChGrp`$(\tilde{\mathcal{G}}, \mathcal{A},$ NeededOverlap$, \varnothing)$
      if *SplitGrp* $\neq \varnothing$ **then**
        SplitRemaining=*SplitGrp* $\cap\, \mathcal{A}$
        $\mathcal{A}$ = *SplitGrp* $\cup\, \mathcal{A}$
        $\mathcal{T}[n_{\mathrm{T}}] = \mathcal{T}[n_{\mathrm{T}}] \cup \{$(*SplitGrp*$\backslash\mathcal{A}$,*SplitGrp*$)\}$
        $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \setminus \{$*SplitGrp*$\}$
    if *SplitRemaining* $= \emptyset$ **then** `%SECTION 2`
      SplitGrp $=$ `ChGrp`$(\tilde{\mathcal{G}}, \varnothing, 0, \varnothing)$
      SplitRemaining $=$ *SplitGrp*
      $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \setminus \{$*SplitGrp*$\}$
    `%SECTION 3`
    $n_{\mathrm{T}} = n_{\mathrm{T}} + 1$
    Set $\mathcal{A}$ to be a group of *SplitRemaining* with
      $|\mathcal{A}| = \min\{$*Residual*, $|$*SplitRemaining*$|\}$
    $\mathcal{T}[n_{\mathrm{T}}] = \{(\mathcal{A}$,*SplitGrp*$)\}$
    SplitRemaining = *SplitRemaining*$\backslash\mathcal{A}$
  WholeGroup=`ChGrp` ($\tilde{\mathcal{G}}$,*SplitGrp*$\backslash\mathcal{A}$,$|$*SplitGrp*$\backslash\mathcal{A}|$,$\mathcal{A}$)
  if *WholeGroup* $\neq \varnothing$ **then** `%SECTION 4`
    $\mathcal{A}$=*WholeGroup* $\cup\mathcal{A}$
    $\mathcal{T}[n_{\mathrm{T}}] = \mathcal{T}[n_{\mathrm{T}}] \cup \{$(*WholeGroup*,*WholeGroup*$)\}$
  else
    if $0 < |\mathcal{A}| < M + 1$ **then** `%SECTION 5`
      $\mathcal{A}$=*SplitGrp*
      $\mathcal{T} = \{$(*SplitRemaining*,*SplitGrp*$)\}$
      SplitRemaining$= \varnothing$
    else
      NewSlot $=$ `True`
  $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \backslash$ *WholeGroup*
  if $M + L - |\mathcal{A}| < M + 1$ **then**
    NewSlot $=$ `True`

Algorithm 1: The main greedy algorithm.

An intuitive explanation for the success of the algirthm is as follows. When the number of users is large, there are several possible choices to fill the allocation of each time block. Hence, the greedy algorithm will almost always succeed to allocate $M + L$ users in each block. A conflict, where there are no available groups that comply with the conditions of Theorem 1, may only occur at the end of the allocation

Function $\mathcal{B} = \texttt{ChGrp}(\mathcal{G}, \mathcal{C}, m, \mathcal{A})$
  %SECTION 6
  $\mathcal{Q} = \{\mathcal{B} \in \mathcal{G} : |\mathcal{B} \cap \mathcal{C}| = m \text{ and } \mathcal{B} \cap \mathcal{A} = \emptyset\}$
  if $|\mathcal{Q}| = 0$ **then**
    $\mathcal{B} = \emptyset$
  else
    $\mathcal{B} = \arg\max_{\mathcal{D} \in \mathcal{Q}} \sum_{a \in \mathcal{D}} \sum_{\mathcal{C} \in \mathcal{G}} |\{a\} \cap \mathcal{C}|$
  end
end

Algorithm 2: The function ChGrp to choose a group.

process. This will yield in erving less than $M + L$ in the last time blocks, and thus a negligible reduction in the DoF.

It is worth mentioing that, for the case of integer α and large number of users, even a random selection of the groups at each stage will result in a close to optimal DoF, as shown in [10]. The extension of this result to non-integer α is possible, but quite hard to track.

V. Numerical results

In this section we show the capability of the presented scheme to achieve a close-to-optimal performance for any combination of M, L, and U. As an example we consider files of 10Mbits, and achievable user rate of $R = 10$Mbps. Thus, with L antennas and without using cache, we can serve all users at U/L seconds.

Fig. 1 depicts the number of slots required to serve all the users, using the greedy algorithm of Section IV, and compares it against the fundamental lower bound in (10). We see that for small number of users, the greedy algorithm may be sub-optimal, and require up to 37% more than the bound (for $M = 3$, $L = 6$ and $U = 9$). Yet, as the number of users increases, the allocation becomes much more efficient, and the required time gets very close to the bound.

Fig. 2 shows the DoF achieved by the greedy algorithm. Again we see that for more than 15 users, the algorithm is very efficient, and the algorithm can achieve at least 97% of $M + L$, which is the ideal DoF (c.f., (11)).

VI. Conclusions

In this paper the sub-packetization problem for a cache-aided MISO communication system is studied. Typically, in a MISO system, the file segments need to be further split beyond the original sub-packetization of Maddah Ali-Niesen scheme [1]. Here, we provided a set of conditions for group allocation that allow all served users to decode their desired file segments without interference. A greedy algorithm is suggested to provide an allocation that satisfies such conditions. While the proposed algorithm has a low-complexity and works with the sub-packetization level of Maddah Ali-Niesen, it performs very close to the optimum performance, and offers a DoF which matches with the fundamental upper bound for a large enough number of users.

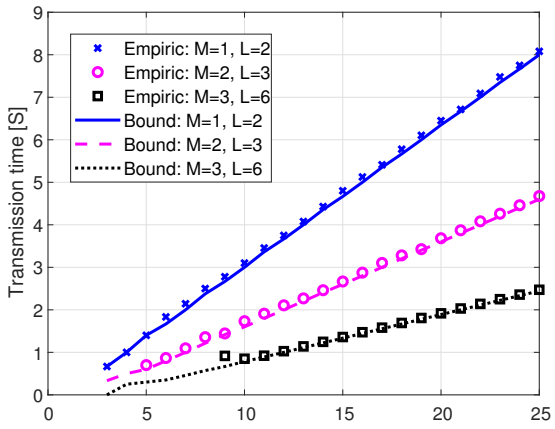

Fig. 1. Transmission time as a function of number of users for various combinations of cache size and number of BS antennas (File size is 10Mbit and user rate is 10Mbps).

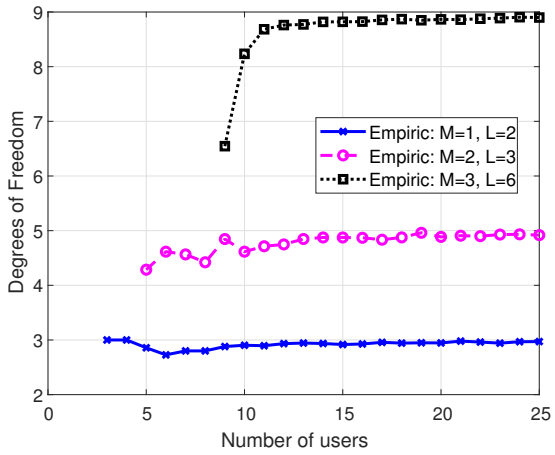

Fig. 2. Number of DoF achieved using the proposed system for various combinations of cache size and number of BS antennas. The upper bound on the DoF is $M + L$.

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[2] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Multi-antenna coded caching," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2113–2117.

[3] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5821–5833, Sept 2017.

[4] S. Jin, Y. Cui, H. Liu, and G. Caire, "Uncoded placement optimization for coded delivery," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2018, pp. 1–8.

[5] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-szeméredi graphs," in *IEEE ISIT*, 2017, pp. 1237–1241.

[6] L. Tang and A. Ramamoorthy, "Low subpacketization schemes for coded caching," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2790–2794.

[7] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1176–1188, 2018.

[8] I. Bergel and S. Mohajer, "Cache aided communications with multiple antennas at finite SNR," *IEEE J. Sel. Areas Commun.*, 2018.

[9] S. Mohajer and I. Bergel, "MISO cache-aided communication with reduced subpacketization," in *Accepted for publication in 2020 IEEE International Conference on Communications (ICC)*, 2020.

[10] ——, "Practical MISO cache-aided communication," *in preparation.*