2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC) | 978-1-6654-9455-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/SPAWC51304.2022.9834007

Channel Aware Greedy Algorithm for MISO Cache-Aided Communication

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

Soheil Mohajer
Dept. of Electrical and Computer Engineering
University of Minnesota, Twin Cities, MN, USA
soheil@umn.edu

***Abstract*—We present novel schemes for cache-aided communication over networks with a multi-antenna base station (BS) that serves multiple single-antenna users. The schemes are based on a greedy scheduling [1], which simultaneously transmits coded messages to disjoint groups of users. The proposed algorithms use the channel state information to opportunistically choose the groups to be served together and to allocate power to each coded message in order to minimize the overall communication delay. Numerical study shows that the new schemes outperform the previously known schemes.**

***Index Terms*—Coded caching, MISO, Subpacketization level, heterogeneous network, greedy algorithm, load balancing.**

I. Introduction

With the overwhelming growth of size of the network as well as the data in content delivery networks, the conventional communication methods seem to be insufficient to provide the desired quality of service. The *coded caching* or *cache-aided communication* [2] has shown a significant potential to increase the overall network throughput. This strategy allows to prefetch (part of) the data at the users' end during the off-peak hours of the network, in order to reduce the traffic at the peak hours. The key feature of coded caching is the fact that caching a packet at one user and requested by another provides an opportunity for *multicasting combined packets* to serve both users simultaneously. This leads to a global gain in addition to the (typically) negligible local gain, which scales with the size of the network. The resulting achievable degrees of freedom (DoF) is proportional to the number of copies of the database cooperatively cached across all the users.

Several issues rise in adoption of coded caching in practical wireless networks. In particular, [3]–[6] studied coded caching in wireless networks in the presence of fading and/or erasure channels. Coded caching for heterogeneous networks with various channels and user rates is studied in [7]. Coded caching in wireless networks with multiple antennas at transmitters and/or receivers is considered in [4], [8]–[12]. Interestingly, the spatial diversity gain and caching gains can be simultaneously achieved. In [9], it is showed that $L + M$ degrees of freedom can be achieved in a broadcast system with L transmit antennas and an aggregate cache that can distributedly store M copies of the database across the users.

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

Subpacketization, i.e., the number of segments each file has to be divided to, is a critical concern in adoption of cache aided communication for practical systems. A large subpacketization level leads to a complex and computationally heavy scheme, where a huge number of short length file segments need to be individually encoded at the transmitter and decoded at the users. The scheme in [9] requires dividing each file into $O\left(U^{M+L-1}\right)$ file segments, where U is the number of users in the network. This is a practically infeasible number, specially when since cache-aided communication is attractive mostly for networks with large number of users.

Several follow-up works focused on lowering this parameter [13]–[16]. In particular, this work is based on [1], in which the subpacketization does not depend on L, and it is only $O\left(U^M\right)$. This can be done by scheduling disjoint groups to be simultaneously served.

In two seminal works, Lampiris and Elia [12] and Salehi et al. [17] much lower subpacketization levels are achieved. The former is only applicable when $L|M$ and is based on the cache replication technique. The latter is applicable for $L \geq M$ and is based on *cyclic* cache placement and transmission scheduling, which leads to subpacketization of only $O(U)$. However, in both cases, an individual stream is sent for each active user and interference cancellation is performed at signal level. This is in contrast with grouping based schemes (e.g., [1], [9]) with bit-level interference cancellation, where one stream is sent to each group of $M + 1$ users, which lead to a power saving factor of $M + 1$, in comparison to signal-level interference cancellation.

In this work, we present two scheduling algorithms for multiple input single output (MISO) cache-aided communication networks, that address the heterogeneity of the network by opportunistically selecting the groups to be simultaneously served. The proposed algorithms are based on [1], but choose the groups to be served in each time slot such that the overall communication delay is reduced. In the low-to-moderate SNR regime, the beamforming gain becomes as crucial as the multiplexing gain, and scheduling groups according to their channel quality offers a significant gain. Furthermore, our algorithms allow to trade the DoF with the beamforming gain.

It is worth noting that the trade-off between the DoF and the beamforming gain in cache-aided communication have been already studied (e.g., in [17], [18]), where the optimal beam

vectors were derived. The distinction of this work is the group scheduling, that allows for trading the gains at each transmission, independently. However, the optimum beamforming derived in [17] can be used on top of the proposed scheduling to further improve the system performance.

In the following, we first present the system model in Section II. The proposed schemes are described in Section III. Finally, we present some numerical simulation in Section IV.

II. System Model

A. The Channel Model

We consider a network with U users each with one receive antenna, and a BS which is equipped with L transmit antennas. The mth received sample after matched filtering at user u is

$$y_u[m] = \mathbf{h}_u \mathbf{x}[m] + z_u[m], \tag{1}$$

where $\mathbf{x}[m] \in \mathbb{C}^{L\times 1}$ is the transmit vector at time m, $z_u[m] \sim \mathbb{C}\mathcal{N}(0, N_0)$ is the additive white Gaussian noise sample at user u and $\mathbf{h}_u \in \mathbb{C}^{1\times L}$ is the channel vector from the BS to user u. We assume the BS has perfect channel state information, and has a total power constraint of $\mathbb{E}\left[\|\mathbf{x}\|^2\right] \le P$.

B. The Placement Strategy

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. In a cache aided communication system, each user $u \in [U]$ is equipped with a memory Z_u that can store up to MNF/U bits. That is, M copies[1] 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. The placement strategy we consider in this work is similar to that of [2]. In particular, we first split each file into $\binom{U}{M}$ segments, and label them with subsets of $[U]$ of size M. More precisely,

$$W_n = \{W_n^{\mathcal{S}} : \mathcal{S} \subseteq [U], |\mathcal{S}| = M\}, \quad n \in [N]. \tag{2}$$

The cache of user $u \in [U]$ will be filled by all file segments whose label contains u. That is,

$$Z_u = \bigcup\nolimits_{n\in[N]} \{W_n^{\mathcal{S}} : u \in \mathcal{S}\}. \tag{3}$$

This implies $|Z_u| = N\binom{U-1}{M-1}F/\binom{U}{M} = NMF/U$, and the cache size constraint is satisfied.

Afterwards, each user requests one file from the dictionary., and we denote the file requested by user u by W_{d_u}. The BS serves all the users during the *delivery phase*, so that each user u is able to decode W_{d_u} from Z_u and the signal received from the BS. The delivery phase is divided into several time slots. In each time slot, we can serve *up to* $M + L$ users. Serving users in each time slot is performed by serving *groups* of size $M + 1$. Hence, the number of groups to be served in each time slot is given by $g \triangleq \frac{M+L}{M+1}$. In this paper, we limit the discussion to the case that g is an integer.

[1]We assume M is an integer, otherwise, a cache-sharing strategy with parameters $\lfloor M \rfloor$ and $\lceil M \rceil$ is needed.

Let $\mathcal{G} \triangleq \{\mathcal{B} \subseteq [U] : |\mathcal{B}| = M + 1\}$ be the collection of all *groups* of size $M + 1$ in $[U]$. In each time slot m, we can serve g groups. In the following, we specify the properties of groups that can be simultaneously served.

Definition 1 (Scheduling for integer g)**.** *For integer value of g, a* scheduling *is a table $\mathcal{T}$ with T rows, where each row $\mathcal{T}[m]$ is a collection of at most g pairs $(\mathcal{B}, \alpha_{\mathcal{B}}[m])$, with $\mathcal{B} \in \mathcal{G}$ and $\alpha_{\mathcal{B}}[m] \in (0, 1]$, satisfying*

(i) All groups are fully served, that is, for every group $\mathcal{B} \in \mathcal{G}$ we have $\sum_{m=1}^{T} \alpha_{\mathcal{B}}[m] \ge 1$.

(ii) All groups in each row are disjoint, i.e., for every m, we have $\mathcal{B} \cap \mathcal{B}' = \varnothing$, for every $\mathcal{B}, \mathcal{B}' \in \mathcal{T}[m]$.

For a time slot $m \in [T]$, the set of *active* users in time slot m is given by $\mathcal{U}[m] = \bigcup_{\mathcal{B}\in\mathcal{T}[m]} \mathcal{B}$. Each user $v \in \mathcal{B} \in \mathcal{T}[m]$ will be served by an $\alpha_{\mathcal{B}}[m]$ fraction of the file segment $W_{d_v}^{\mathcal{B}\setminus\{v\}}$ during the time slot m. To this end, we form a coded file segment for the group $\mathcal{B}$, given by

$$W_{\mathcal{B}} \triangleq \bigoplus_{u\in\mathcal{B}} W_{d_u}^{\mathcal{B}\setminus\{u\}}. \tag{4}$$

We modulate the corresponding $\alpha_{\mathcal{B}}[m]$ fraction of this coded file segment to a codeword $\tilde{W}_{\mathcal{B}}[m]$, that consists of τ channel symbols. The transmit signal for time slot m is given by

$$\mathbf{X}[m] = \sum_{\mathcal{B}\in\mathcal{T}[m]} \sqrt{P_{\mathcal{B}}}\mathbf{v}_{\mathcal{B}}\tilde{W}_{\mathcal{B}}[m], \tag{5}$$

where $\mathbf{v}_{\mathcal{B}}$ is the beamforming vector for group $\mathcal{B}$ (with $\|\mathbf{v}_{\mathcal{B}}\| = 1$), and $\sqrt{P_{\mathcal{B}}}$ is the power allocated to group $\mathcal{B}$, satisfying $\sum_{\mathcal{B}\in\mathcal{T}[m]} P_{\mathcal{B}} \le P$. Lastly, the ℓth row of $\mathbf{X}[m] \in \mathbb{C}^{L\times\tau}$ is sent over the ℓth transmit antenna at time slot m.

C. The beamforming vectors

The optimal beam vectors were recently derived in [17], [18]. Here, we adopt a sub-optimum, yet simpler, zero-forcing approach that allows focusing on the scheduling. Note that, for an optimization set of groups to be simultaneously served, the optimal beam vectors can be also used in conjunction with the proposed schemes, to further improve the performance.

For a given time slot m with $\mathcal{U}[m] = \bigcup_{\mathcal{B}\in\mathcal{T}[m]} \mathcal{B}$ being the set of active users, we choose the beam vector for group $\mathcal{B}$ as

$$\mathbf{v}_{\mathcal{B}} = \underset{\substack{\mathbf{v}:\|\mathbf{v}\|=1 \\ \mathbf{h}_j^T\mathbf{v}=0, \forall j\in\mathcal{U}[m]\setminus\mathcal{B}}}{\arg\max} \; \min_{i\in\mathcal{B}} |\mathbf{h}_i^T\mathbf{v}|. \tag{6}$$

If $|\mathcal{T}[m]| = g$, then, with high probability, the optimization problem in (6) has a single solution, since $\mathbf{v} \in \mathbb{C}^L$ should be orthogonal to $(g-1)(M+1) = L-1$ directions. However, when $|\mathcal{T}[m]| < g$, we have some freedom in the choice of the beam, and its optimization typically leads to higher gains.

Let $H_m^T \in \mathbb{C}^{L\times|\mathcal{U}[m]|}$ be a matrix obtained by stacking the channel vectors of all active users in time slot m. The second constraint in (6) can be written as $H_m\mathbf{v}_{\mathcal{B}} = E_{\mathcal{B}}\tilde{\mathbf{d}}_{\mathcal{B}}$ where $E_{\mathcal{B}}$ is a binary matrix with $|\mathcal{U}[m]|$ rows indexed by the users in $\mathcal{U}[m]$ and $|\mathcal{B}|$ columns, labeled by the users in $\mathcal{B}$, and

$E_{u,v} = 1$ if and only if $u = v$. With this formulation, we wish to maximize the minimum entry of $\tilde{\mathbf{d}}_{\mathcal{B}}$.

The problem can be further simplified by defining

$$\gamma_{\mathcal{B}} = \min_{i \in \mathcal{B}} |\mathbf{h}_i^T \mathbf{v}_{\mathcal{B}}|, \tag{7}$$

and normalized vectors $\mathbf{u}_{\mathcal{B}} = \mathbf{v}_{\mathcal{B}}/\gamma_{\mathcal{B}}$ and $\mathbf{d}_{\mathcal{B}} = \tilde{\mathbf{d}}_{\mathcal{B}}/\gamma_{\mathcal{B}}$. Then, the problem in (6) can be rephrased as

$$\mathbf{u}_{\mathcal{B}}, \mathbf{d}_{\mathcal{B}} = \underset{\substack{\mathbf{u},\mathbf{d}:|d_i|\geq 1 \\ H_m \mathbf{u} = E_{\mathcal{B}} \mathbf{d}}}{\arg\min} \|\mathbf{u}\|^2. \tag{8}$$

It is easy to show that $\mathbf{v}_{\mathcal{B}}$ should be in the column-span of H_m, i.e., $\mathbf{u}_{\mathcal{B}} = H_m^T \mathbf{q}$, for some $\mathbf{q} \in \mathbb{C}^{|\mathcal{U}[m]|}$. Combining this with $H_m \mathbf{u}_{\mathcal{B}} = E_{\mathcal{B}} \mathbf{d}_{\mathcal{B}}$, we conclude that $\mathbf{u}_{\mathcal{B}} = H_m (H_m H_m^T)^{-1} E_{\mathcal{B}} \mathbf{d}_{\mathcal{B}}$, where $\mathbf{d}_{\mathcal{B}}$ is the solution of

$$\mathbf{d}_{\mathcal{B}} = \underset{\mathbf{d}:|d_i|\geq \mathbf{1}}{\arg\min} \mathbf{d}^T E_{\mathcal{B}}^T (H_m H_m^T)^{-1} E_{\mathcal{B}} \mathbf{d}. \tag{9}$$

Defining $D_{\mathcal{B}} = \mathbf{d}_{\mathcal{B}} \mathbf{d}_{\mathcal{B}}^T$, we can rewrite the problem in (8) as

$$D_{\mathcal{B}} = \underset{\substack{D:\mathsf{diag}(D)\geq \mathbf{1} \\ D \succeq 0 \\ \mathsf{rank}(D)=1}}{\arg\min} \mathrm{Tr}(E_{\mathcal{B}}^T (H_m H_m^T)^{-1} E_{\mathcal{B}} D). \tag{10}$$

Using semi-definite relaxation [19], we drop the rank constraint, and efficiently solve the resulting convex problem.

III. The Proposed Schemes

In this section, we propose two greedy algorithms for the scheduling mechanism. The scheduling algorithms are generalizations of the greedy scheduling presented in [1]. When U is large, there is a numerous possible schedules with the minimum number of time-slots. The overall delay depends on the set of groups that are simultaneously served. In particular, the number of channel symbols in each time slot depends on the channel gains of all active groups. The goal is to schedule the groups to minimize the overall delay. However, this leads to an infeasible optimization problem consisting of group selection, beam forming evaluation, and power allocation. Instead, we present two heuristic algorithms.

A. Channel-Aware Scheduling

In order to improve the performance, we propose a greedy scheduling mechanism to minimize the duration of each time slot. At each stage m, we start $\mathcal{T}[m]$ with an unserved group with the minimum total number of served packets, and iteratively add groups to $\mathcal{T}[m]$. Here, we assume $\alpha_{\mathcal{B}}[m] = 1$ for every $\mathcal{B} \in \mathcal{T}[m]$. We also determine the power allocated to the active groups, so that they all complete the transmission at the same time. For this we need an identical $P_{\mathcal{B}} \gamma_{\mathcal{B}}^2$ for all $\mathcal{B} \in \mathcal{T}[m]$. This together with $\sum_{\mathcal{B} \in \mathcal{T}[m]} P_{\mathcal{B}} = P$ imply

$$P_{\mathcal{B}} = \frac{P}{\gamma_{\mathcal{B}}^2 \sum_{\mathcal{C} \in \mathcal{T}[m]} 1/\gamma_{\mathcal{C}}^2}, \tag{11}$$

leading to a common rate at time slot m, given by

$$R_m = \log_2 \left(1 + \frac{P}{\sum_{\mathcal{C} \in \mathcal{T}[m]} 1/\gamma_{\mathcal{C}}^2} \right). \tag{12}$$

Algorithm 1: The channel-aware greedy algorithm.

Input: Parameters M, L, U, $\{\mathbf{h}_u : u \in [U]\}$.
Output: Collision-Free Scheduling $\mathcal{T}$.

Initialization:
  $\tilde{\mathcal{G}}$ = All Subsets of $[U]$ of Size $M+1$
  $\mathsf{Scr}(x) = \mathbf{0}, \ \forall x \in [U]$
  $m = 0$
  $\mathcal{T}$ = An empty array;
while $|\tilde{\mathcal{G}}| > 0$ **do**
  $m = m + 1$, $\mathcal{U} = \varnothing$, $\mathcal{B} = \{1\}$
  while $|\mathcal{U}| \leq M + L$ *and* $\mathcal{B} \neq \varnothing$ **do**
    $\mathcal{Q} = \{\mathcal{X} \in \tilde{\mathcal{G}} : \mathcal{X} \cap \mathcal{U} = \varnothing\}$
    if $\mathcal{U} = \varnothing$ **then**
      $\mathcal{B} = \arg\min_{\mathcal{X} \in \mathcal{Q}} \sum_{x \in \mathcal{X}} \mathsf{Scr}(x)$
    else
      $\mathcal{B} = \arg\min_{\mathcal{X} \in \mathcal{Q}} \sum_{\mathcal{C} \in \mathcal{A} \cup \mathcal{X}} \sum \frac{1}{\gamma_{\mathcal{C}}^2}$
    if $\mathcal{B} \neq \varnothing$ **then**
      $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \setminus \{\mathcal{B}\}$
      for $x \in \mathcal{B}$ **do**
        $\mathsf{Scr}(x) = \mathsf{Scr}(x) + 1$
    $\mathcal{T}[m] = \mathcal{T}[m] \cup \{\mathcal{B}\}$
    $\mathcal{U} = \mathcal{U} \cup \mathcal{B}$

The new groups will be added according to the following greedy selection criteria:

(i) compute the beamforming vectors for each remaining group, as in (6)-(10);
(ii) for the beamforming vectors obtained in (i), derive the power allocation from (11), so that all the groups achieve the same rate;
(iii) choose the group that maximizes the common group rate in (12), and add it to $\mathcal{T}[m]$.

This procedure is formally presented in Algorithm 1.

B. Scheduling with Packet splitting

When SNR is low, allocation of power to achieve a common rate for all the active groups (and hence completely serve the coded packets of all groups) can be very inefficient. The bottleneck is in groups, with very weak combination of channel condition and beamforming vector (i.e., $\gamma_{\mathcal{B}}$ in (7)), for which we need to allocate a large fraction of the transmit power to compensate for the channel gain. In contrast, we can assign a lower rate to such groups, and only serve part of the desired coded packet. In return, the rest of the packet has to be served in other time slot(s), when the group is scheduled with other groups, and perhaps has a better beamforming vector.

We start with the scheduling mechanism described in Section III-A. However, once the groups in $\mathcal{T}[m]$ are determined, we allow for splitting the packets into parts (i.e., $\alpha_{\mathcal{B}}[m] \leq 1$). While there are numerous possibilities of splitting the packets, we need to limit the set of options so that we can analytically track them. To this end, we formulate an optimization problem to determine $\alpha_{\mathcal{B}}[m]$ and the transmit power for each packet.

We denote by $F_{\mathcal{B}}$ the size of the remaining part of the packet intended for group $\mathcal{B}$. Initially, we have $F_{\mathcal{B}} = |W_{\mathcal{B}}| = F/\binom{U}{M}$ for every group $\mathcal{B}$. As we proceed through serving the groups, the size of the remaining part of the packets reduce.

Consider a time slot m at which groups in $\mathcal{T}[m]$ are scheduled to be served. If we only serve $\alpha_{\mathcal{B}}[m]$ fraction of $W_{\mathcal{B}}$ jointly and serve the remaining $(1-\alpha_{\mathcal{B}}[m])$ fraction individually, the overall delay will be

$$\tau(\boldsymbol{\alpha}) = \sum_{\mathcal{B}\in\mathcal{T}[m]} \frac{(1-\alpha_{\mathcal{B}}[m])F_{\mathcal{B}}}{\log_2(1+\tilde{\gamma}_{\mathcal{B}}^2 P)} + \max_{\mathcal{B}\in\mathcal{T}[m]} \frac{\alpha_{\mathcal{B}}[m]F_{\mathcal{B}}}{\log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}})},$$

where $\tilde{\gamma}_{\mathcal{B}}$ is the gain for group $\mathcal{B}$ when it is served individually, and $\boldsymbol{\alpha} = (\alpha_{\mathcal{B}}[m] : \mathcal{B} \in \mathcal{T}[m])$. Clearly, $\tau(\boldsymbol{\alpha})$ is minimized when all the terms in the maximization term are identical. Denoting the duration of jointly serving by T_0, we obtain

$$\alpha_{\mathcal{B}}[m] = T_0 \log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}})/F_{\mathcal{B}}. \tag{13}$$

Thus, minimizing $\tau(\boldsymbol{\alpha})$ with respect to $\{P_{\mathcal{B}} : \mathcal{B} \in \mathcal{T}[m]\}$ and T_0 will be equivalent to

$$\begin{aligned} \text{Minimize} \quad & \sum_{\mathcal{B}\in\mathcal{T}[m]} \frac{F_{\mathcal{B}} - T_0 \log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}})}{\log_2(1+\tilde{\gamma}_{\mathcal{B}}^2 P)} + T_0 \\ \text{Subject to} \quad & P_{\mathcal{B}} \geq 0 \\ & \sum\nolimits_{\mathcal{B}\in\mathcal{T}[m]} P_{\mathcal{B}} \leq P \\ & T_0 \leq F_{\mathcal{B}}/\log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}}) \quad \forall \mathcal{B} \in \mathcal{T}[m]. \end{aligned} \tag{14}$$

Taking derivative with respect to $P_{\mathcal{B}}$, we arrive at

$$P_{\mathcal{B}} = \max\left\{0, \frac{\lambda}{\log_2(1+\tilde{\gamma}_{\mathcal{B}}^2 P)} - \frac{1}{\gamma_{\mathcal{B}}^2}\right\}, \tag{15}$$

which provides us with the optimal power allocation, using a variant of the water-filling algorithm. The constant λ can be evaluated from $\sum_{\mathcal{B}\in\mathcal{T}[m]} P_{\mathcal{B}} = P$.

After we have the optimal values for $P_{\mathcal{B}}$, we have a linear function with respect to T_0 to be minimized. Clearly, the optimum T_0 is one of two extreme values, i.e., we either have

$$T_0 = 0 \quad \text{or} \quad T_0 = \min_{\mathcal{B}\in\mathcal{T}[m]} \frac{F_{\mathcal{B}}}{\log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}})}. \tag{16}$$

It is worth mentioning that, even though the delay $\tau(\boldsymbol{\alpha})$ is evaluated based on joint and individual serving of the groups, the delivery in time slot m will be as follows: if $T_0 = 0$, then we only serve the first selected group; otherwise, all the groups (with positive $P_{\mathcal{B}}$) will be jointly served. In either case, we keep the remaining part of each packet for future time slots. Hence, for $T_0 > 0$, we update $F_{\mathcal{B}}$ as

$$F_{\mathcal{B}} \leftarrow F_{\mathcal{B}} - T_0 \log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}}),$$

for every $\mathcal{B} \in \mathcal{T}[m]$. This immediately implies that at least one group will be completely served at time slot m, and hence, the number of required time slots is upper bounded by the number of groups. The scheduling mechanism described above is formally presented in Algorithm 2.

Algorithm 2: The packet-splitting algorithm

Input: Parameters M, L, U, $\{\mathbf{h}_u : u \in [U]\}$,$P$.
Output: Collision-Free Scheduling $\mathcal{T}$ and power allocation P

Initialization:
  $\tilde{\mathcal{G}}$ = All subsets of $[U]$ of Size $M+1$
  $\mathsf{Scr}(x) = \mathbf{0} \quad \forall x \in [U]$
  $m = 0$
  $F_{\mathcal{B}} = F/\binom{U}{M} \quad \forall \mathcal{B} \in \tilde{\mathcal{G}}$
while $|\tilde{\mathcal{G}}| > 0$ **do**
  $m = m+1$, $\mathcal{U} = \varnothing$, $\mathcal{B} = \{1\}$, $\mathcal{T}[m] = \{\ \}$
  while $|\mathcal{U}| < L+M$ *and* $\mathcal{B} \neq \varnothing$ **do**
    $\mathcal{Q} = \{\mathcal{X} \in \mathcal{G} : \mathcal{X} \cap \mathcal{U} = \varnothing\}$
    if $\mathcal{U} = \varnothing$ **then**
      $\mathcal{B}, \mathcal{B}_0 = \arg\min_{\mathcal{X}\in\mathcal{Q}} \sum_{x\in\mathcal{X}} \mathsf{Scr}(x)$
    else
      $\mathcal{B} = \arg\min_{\mathcal{X}\in\mathcal{Q}} \sum_{\mathcal{C}\in\mathcal{U}\cup\mathcal{X}} \sum \frac{1}{\gamma_{\mathcal{C}}^2}$
    $\mathcal{T}[m] = \mathcal{T}[m] \cup \{\mathcal{B}\}$
    $\mathcal{U} = \mathcal{U} \cup \mathcal{B}$
  Calculate optimal powers using (15) and (14)
  Calculate T_0 using (16)
  if $T_0 > 0$ **then**
    for $\mathcal{B} \in \mathcal{T}[m]$ **do**
      $F_{\mathcal{B}} = F_{\mathcal{B}} - T_0 \log_2(1+\gamma_{\mathcal{B}}^2 P_{\mathcal{B}})$
      if $F_{\mathcal{B}} = 0$ **then**
        $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \setminus \mathcal{B}$
        for $x \in \mathcal{B}$ **do** $\mathsf{Scr}(x) = \mathsf{Scr}(x) + 1$
  else
    $\mathcal{T}[m] = \{\mathcal{B}_0\}$
    $F_{\mathcal{B}_0} = 0$
    $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \setminus \{\mathcal{B}_0\}$
    for $x \in \mathcal{B}_0$ **do** $\mathsf{Scr}(x) = \mathsf{Scr}(x) + 1$

IV. Numerical results

We conduct Monte Carlo simulations to compare the proposed algorithms against other scheduling methods. We consider a network with $U = 15$ users, each storing $2/15$ of the database in its cache (i.e., $M = 2$), and a BS with $L = 4$ antennas. Thus, we have groups of size $M+1 = 3$ users and (up to) $g = \frac{M+L}{M+1} = 2$ groups can be simultaneously served. Our performance metric is the total delay of serving all users with files of size $F = 10^6$ bits, over a channel with a bandwidth of 1 MHz. This delay is plotted as a function of SNR, which is defined as $P \cdot \mathbb{E}\left[|(\mathbf{h}_u)_i|^2\right]/N_0$ for user u.

Fig. 1 corresponds to a *homogeneous* scenario, where all the fading channels are distributed as $\mathbf{h}_u \sim \mathbb{CN}(\mathbf{0}, \mathbf{I})$. We compare the performance of the two proposed algorithms (referred to as "Ch aware" and "Packet split", respectively), against the "Greedy" algorithm of [1] (which selects the groups only based on the number of the total number of packets left for all the users in the group). It is shown in [1] that the performance of the "Greedy" algorithm is very similar to that of [9], in spite of having a much lower subpacketization.

The "Greedy" algorithm performs close to optimal at high SNR, where the multiplexing gain dominates the beamforming gain. We also implement the "No ZF" algorithm, in which only $g = 1$ group is served at a time, using only the cache data to suppress the interference (as in Section III of [9], but with optimal power allocation.) Clearly, the "No ZF" algorithm achieves the maximal beamforming gain for each active group, and hence has the best performance at low SNR.

Inspecting the channel aware algorithm, we observe a significant improvement compared to the "Greedy" algorithm for the entire range of SNR. However, the channel aware still looses to the "No ZF" algorithm at low-to-moderate SNR, since it still aims at jointly serving $g = 2$ groups. While one would

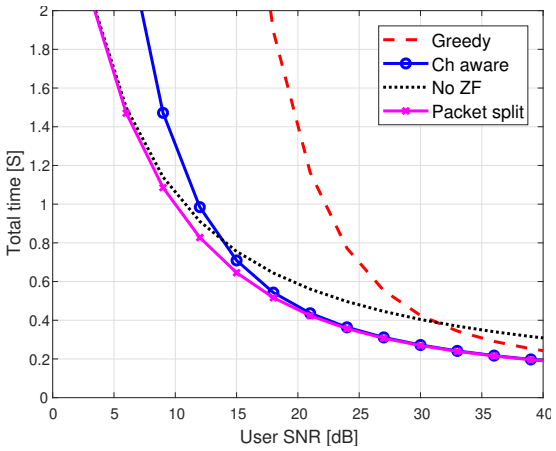

Fig. 1: Transmission time vs. users' SNR in a homogeneous network with $(U, L, M) = (15, 4, 2)$.

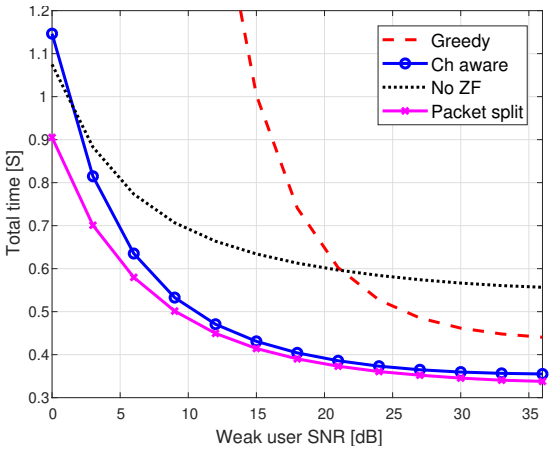

Fig. 2: Transmission time vs. the weak user's SNR in the presence of 14 users with SNR= 36dB, $(U, L, M) = (15, 3, 1)$.

expect the beamforming gain to be negligible at high SNR, the figure demonstrates that this actually happens only at very high SNR. For example, even at SNR of 40dB, the Channel aware algorithm requires 30% less time (or alternatively more than 5dB less power) compared to the "Greedy" algorithm. This gain comes solely from the better selection of groups to join, which leads to higher beamforming gains.

The packet splitting algorithm simultaneously achieves the performance of the channel aware algorithm at high SNR and the "No ZF" algorithm at low SNR. This algorithm optimizes the power allocation per packet at the actual operating SNR. Thus, it will jointly transmit multiple packets at high SNR, but converges to the "No ZF" performance at low SNR. In the best case, the Packet splitting algorithm is more than 1dB better than the minimum of the "Greedy" and the "No ZF" algorithms. It is also worth noting that the additional complexity of packet splitting is negligible (as the power allocation is solved almost in closed form).

To further demonstrate the advantage of packet splitting we consider a system with parameters $(U, L, M) = (15, 3, 1)$, where 14 users experience strong channels (with SNR of 36dB) and one user has a weak channel with variable SNR. The weak user requires longer transmission times and hence it is advantageous to schedule its packets over multiple time slots, with various other users. Fig. 2 shows that the packet splitting algorithm significantly outperforms both the "Greedy" and the "No ZF" algorithms. To elaborate the difference, we note that, while there are only 14 groups that include the weak user, at the lowest SNR these groups are scheduled in 42 out of the 105 time slots. That is, each of the 14 coded packets intended for groups including the weak user are broken into smaller parts, each part is jointly transmitted with a whole packet intended for another group. Thus, the packet splitting algorithm is able to better balance this non-homogeneous scenario.

REFERENCES

[1] I. Bergel and S. Mohajer, "Practical scheme for miso cache-aided communication," in *IEEE Workshop Signal Process. Adv. Wirel. Commun. (SPAWC)*, 2020.

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

[3] M. Gregori, J. Gómez-Vilardebò, J. Matamoros, and D. Gündüz, "Joint transmission and caching policy design for energy minimization in the wireless backhaul link," in *IEEE Int. Symp. Inf. Theory*, 2015.

[4] S. Yang, K.-H. Ngo, and M. Kobayashi, "Content delivery with coded caching and massive mimo in 5g," in *Int. Symp. Turbo Codes Iterative Inf. Process (ISTC)*, 2016, pp. 370–374.

[5] S. S. Bidokhti, M. Wigger, and R. Timo, "Noisy broadcast networks with receiver caching," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 6996–7016, 2018.

[6] A. Ghorbel, M. Kobayashi, and S. Yang, "Content delivery in erasure broadcast channels with cache and feedback," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6407–6422, 2016.

[7] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Optimization of heterogeneous caching systems with rate limited links," in *IEEE Int. Conf. Comm. (ICC)*, 2017.

[8] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7253–7271, 2016.

[9] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Multi-antenna coded caching," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2113–2117.

[10] K.-H. Ngo, S. Yang, and M. Kobayashi, "Scalable content delivery with coded caching in multi-antenna fading channels," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 1, pp. 548–562, 2018.

[11] I. Bergel and S. Mohajer, "Cache aided communications with multiple antennas at finite SNR," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1682–1691, 2018.

[12] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1176–1188, 2018.

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

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

[15] 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.

[16] L. Tang and A. Ramamoorthy, "Low subpacketization schemes for coded caching," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2790–2794.

[17] M. J. Salehi, E. Parrinello, S. P. Shariatpanahi, P. Elia, and A. Tölli, "Low-complexity high-performance cyclic caching for large miso systems," *IEEE Trans. Wirel. Commun.*, 2021.

[18] A. Tölli, S. P. Shariatpanahi, J. Kaleva, and B. H. Khalaj, "Multi-antenna interference management for coded caching," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 3, pp. 2091–2106, 2020.

[19] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. signal process.*, vol. 54, no. 6, pp. 2239–2251, 2006.