MDS Coded Cooperative Caching for Heterogeneous Small Cell Networks

Jialing Liao, Kai-Kit Wong, *Fellow, IEEE*, Yangyang Zhang, Zhongbin Zheng, and Kun Yang

***Abstract*—In this paper, a typical cache-enabled small cell network under heterogeneous file and network settings is considered, where the neighboring base stations are enabled to collaborate to share the cached content. In particular, maximum distance separable (MDS) codes are used for content restructuring in order to take full usage of the content diversity. We aim to minimize the *long-term average* user attrition (UA) cost for fetching content from external storage subject to the overall cache capacity constraint by jointly optimizing the content placement and the cooperative policy throughout the network using the file popularity information. To further reduce the backhual load, a compound caching technique taking the advantages of multicast content delivery and cooperative content sharing, which is referred to as multicast-aware cooperative caching, is then developed. Mathematical analysis and simulation results are presented to illustrate the advantages of MDS codes and cooperative caching in terms of reducing the backhaul requirements.**

***Index Terms*—Caching, cooperation, heterogeneous networks.**

I. Introduction

Caching at the wireless edge is expected to play an important role in the emerging fifth-generation (5G) wireless communication networks as the tremendous mobile traffic growth has imposed high demands on content delivery [1]. By storing popular content at the network edge, the cached content can be delivered to users from local caches rapidly instead of being downloaded from the core network via backhaul which helps to reduce the peak-time traffic and latency.

Currently, considerable research has been done on physical layer caching. Of relevance to our work are [2]–[6] where they focused on the optimization of content placement for cache-enabled small-cell networks. In [2], MDS coded caching was considered with homogeneous network settings which gave rise to identical content placement in all cells. Any cache miss was dealt with by using separate unicast transmissions via the backhaul. In another work [3], the authors studied uncoded multicast-aware caching in delay tolerant networks. The heuristic nature of the proposed methods also makes them not applicable for coded caching scenarios. In our previous work [4], coded multicast-aware caching was proposed in the partly heterogeneous settings of distinct cache and file sizes and homogeneous file popularity and numbers of users in all the cells. While the works mentioned above are offline non-cooperative schemes with limited cache sizes, [5] devised an online cooperative caching scheme with infinite cache capacity. Since the current user demands were given and the caching policies for different files were mutually independent, the formulated problem was actually linear and therefore could be easily solved heuristically. Finally in [6], an in-network cooperative caching scheme was proposed with the strong assumption of knowing the actual file demands of each user as well. Furthermore, the decentralized heuristic scheme is suboptimal, and the heterogeneity of the locations of the SBSs and file popularity in different cells was not well addressed.

To conclude, considering the heterogeneity of cache-enabled small-cell networks, such as distinct file popularity, file sizes, cache sizes, coverages and locations of different SBSs, not only requires redesign of content placement but also cache size allocation amongst the SBSs, as mentioned in [7], [8]. In this setup, cache decomposition in different cells will be generally not the same. Considering also the fact that file sizes may be large compared to the limited cache size in practice, files are usually split into fragments to make full use of the content diversity. However, note that all of the above-mentioned works considered whole file caching except [2], [4]. When the fragments are randomly selected and stored in the caches without coding, both the number of fragments in each cell and which fragments that are stored (i.e., the degree of content duplication amongst the cells), determine the backhaul load. As a result, it would be very difficult for the macro base station (MBS) to deliver the uncached content via a shared link to all the cells and unicast content delivery is therefore commonly used between the MBS and SBSs at the expense of high backhaul cost [2], [5], [6]. On the other hand, cache content overlap among different cells would restrain cooperative caching from being effective.

In this paper, our aim is to unleash the potential of cooperative caching and multicast-aware content delivery by taking advantages of the inherent independence amongst the MDS coded packets for minimizing the average UA cost. In summary, this paper has made the following major contributions:

- We develop offline caching schemes optimizing the long-term average performance by estimating all possible joint user requests utilizing the file popularity information.
- The heterogeneity of the parameters that affects the cache management and cooperative policy is all considered with the coordination among different SBSs and files.
- Furthermore, we derive the performance gains of storing coded packets over uncoded fragments in the caches and

J. Liao and K. K. Wong are with the Department of Electronic and Electrical Engineering, University College London, United Kingdom (e-mail: jialing.liao.14@ucl.ac.uk, kai-kit.wong@ucl.ac.uk).

Y. Zhang is with Kuang-Chi Institute of Advanced Technology, China.

Z. Zheng is with East China Institute of Telecommunications, China Academy of Information and Communications Technology, Shanghai, China.

K. Yang is with University of Electronic Science and Technology of China, Chengdu, Sichuan, China.

This work is supported by EPSRC under grant EP/K015893/1.

quantify the advantages of cooperative caching schemes over common caching schemes.

II. System Model

A. Network Model

A small cell network is considered which comprises a single MBS, and K non-overlapping small cells each consisting of a single SBS and I_k users, for the kth cell. Let $\mathcal{K} \triangleq \{1, \dots, K\}$ denote the set of SBSs which operate in disjoint subchannels with the MBS in order to remove the impact of interference. Besides, any interference among neighboring SBSs is assumed eliminated by techniques such as enhanced inter-cell interference coordination (eICIC) or/and orthogonal multiple access [9], [10]. Each SBS has its own local storage. We assume that the MBS has access to all files in the set $\mathcal{F} \triangleq \{f_1, f_2, \dots, f_N\}$ with respective file sizes $\mathbf{s} \triangleq [s_1, s_2, \dots, s_N]$ while the SBSs have limited cache capacities that are subject to a network-wide total cache capacity budget M. We let M_k denote the cache capacity for SBS k, with $M_k \leq \sum_{j=1}^{N} s_j$. SBSs can push the cached packets to the users when requested while the uncached parts have to be delivered to the SBSs via backhaul from the MBS or cooperative SBSs. Note that the users located outside of any small cells can only be served by the MBS and hence are ignored.

1) Cooperative caching: As shown in Fig. 1, neighboring SBSs can be connected to each other via high-capacity links to share their cached content in different cells collaboratively. In this scheme, the uncached content can be fetched from not only the MBS via backhaul but also the cooperative SBSs via the fronthaul links. Considering the different costs for fetching content from the MBS and the neighboring SBSs, we adopt the concept of user attrition (UA) cost introduced in [5] to evaluate the performance of the cooperative caching scheme.[1] Cache content placement and the policy for SBS cooperation are to be jointly optimized to minimize the UA cost. Unless stated otherwise, this scheme uses unicast for content delivery.

2) Multicast-aware cooperative caching: In this approach, multicast-based content delivery and content sharing amongst neighboring SBSs are combined with the aid of MDS codes. In contrast to conventional cooperative caching, multicasting is applied by the MBS to deliver content to the SBSs requesting the same file simultaneously, see Case II of Fig. 1.

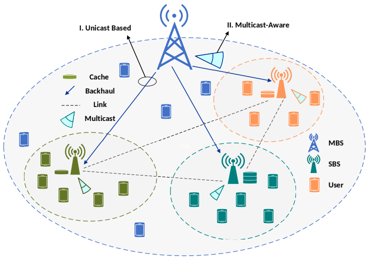


Fig. 1: Cache-enabled heterogeneous small-cell networks.

B. MDS Coding

MDS codes are employed to construct pieces of a file that can be put back together to recover the file. They are particularly suitable for our settings of multicast-aware caching and cooperative caching in which the cached content in different cells needs to be coordinated. Compared to the case of storing uncoded fragments, MDS codes bring a unique benefit that the coded packets are all independent from each other so that a certain number of randomly drawn packets will be sufficient to recover the file. This allows us to use only the number of packets stored in each cell, instead of the details of the packets, to derive the backhaul load, simplifying the analysis.

We parametrize MDS codes by (l_j, n_j) such that file j is cut into n_j fragments and then coded into l_j independent packets by MDS. Any n_j packets can rebuild the entire file.

Considering that the kth SBS caches $m_{k,j}$ coded packets of file j, we let $\mathbf{m}^j = [m_{1,j}, m_{2,j}, \dots, m_{K,j}]$ be the content placement vector for file j. For unicast and multicast-aware cooperative caching scenarios, the total number of packets has to be at least

$$l_j = \sum_{k=1}^{K} m_{k,j} + n_j - \min_{k \in \{1,\dots,K\}} \sum_{t=1}^{K} x_{k,j}^t,$$

where $x_{k,j}^t$ denotes the number of packets delivered from SBS t to SBS k to serve the requests for file j so that there is no content overlap in both content sharing process amongst the cooperative SBSs and content delivery phase at the MBS.

C. File Popularity Profile

Note that users in different cells may have different preferences towards the files. The most popular file in one cell may receive least attentions from another cell. It is thus better to consider local file popularity in each cell rather than the global popularity in the entire network which is often the case in the literature. Without loss of generality, here we assume that the file popularity in each cell obeys Zipf's distribution but with unique skewness parameter and popularity rank. According to the Zipf's law, the frequency for file j to be requested by each user in cell k can then be written as [11]

$$p_{k,j} = \frac{\left(1/\lambda_{k,j}^{\gamma_k}\right)}{\sum_{i=1}^{N} \left(1/i^{\gamma_k}\right)}, \ \forall k, j, \tag{1}$$

where γ_k is the skewness in cell k reflecting the concentration of the popularity distribution and $\lambda_{k,j}$ denotes the rank of the popularity of file j in cell k. For instance, $\lambda_{k,j} = 1$ means file j is the most popular file in cell k. Hence, the probability of file j not being requested by the users in cell k is

$$\alpha_{k,j} = (1 - p_{k,j})^{I_k}, \ \forall k, j. \tag{2}$$

Thus, the probability for file j being requested by at least one of the users in cell k will be $1 - \alpha_{k,j}$.

III. Cooperative Caching

In this section, we consider that the SBSs can fetch content from the neighboring SBSs via some high capacity links and study the optimal cooperative caching policy among the SBSs.

[1] UA cost is the overall cost for fetching content from an external storage.

Note that the independence amongst the MDS coded packets cached in all the cells almost surely guarantees that the shared contents are always non-overlapping.

Cooperative caching consists of three phases:

(i) the content placement phase,
(ii) the cnt sharing phase among the SBSs, and
(iii) the content delivery phase from the MBS via backhaul.

Note that in the content delivery phase, we assume that unicast is used by the MBS to sent uncached content to the SBSs.

To further eliminate the redundancy, we assume that the SBSs can selectively deliver part of the packets from their own caches to the requested SBS rather than thontee whole of the cached packets. The amounts of shared content among the cooperative SBSs are defined as $\mathbf{X} = \{x^t_{k,j}\}_{K\times N\times K}$ where $x^t_{k,j}$ denotes the number of packets delivered from SBS t to SBS k for file j. Accordingly, we let f^t_k be the associated cost when SBS k fetches content from SBS t and f^M_k denote the cost for delivering content to SBS k from the MBS.

The UA costs are proportional to the square of the minimum distances between the associated BSs with the unit cost coefficients defined as f_0 and f^M_0, respectively. Note that $\{f^t_k\}$ must satisfy the triangle inequality, i.e., $f^t_k \le f^t_l + f^l_k$, and the cost for fetching content from local storage can be ignored, i.e., $f^k_k = 0, \forall k$. Moreover, the UA costs for fetching content from the MBS via backhaul are usually higher than those caused by the cooperation between the SBSs due to proximity.

Instead of focusing on the backhaul load in the non-cooperative caching scenarios, our objective here is to minimize the average UA cost, i.e., the cost of fetching content from external storage, subject to a given overall cache capacity constraint by optimizing the cache content placement and cooperation policy jointly. In this case, the expected UA cost defined as $C^{\text{MDS}}_{\text{coop}}$ can be written as

$$C^{\text{MDS}}_{\text{coop}} = \sum_{j=1}^{N}\sum_{k=1}^{K}\left[\left(1-\min\left(1,\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}\right)\right)f^M_k + \sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}f^t_k\right]s_j(1-\alpha_{k,j}). \quad (3)$$

Hence, the problem of interest is given by

$$\min_{\{m_{k,j}\},\{x^t_{k,j}\}} \quad C^{\text{MDS}}_{\text{coop}} \quad (4a)$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\sum_{j=1}^{N}\frac{m_{k,j}}{n_j}s_j \le M, \quad (4b)$$

$$0 \le m_{k,j} \le n_j,\ \forall k,j, \quad (4c)$$

$$0 \le x^t_{k,j} \le m_{t,j},\ \forall k,j,t, \quad (4d)$$

where the cache size allocation problem is merged into the optimization of the content placement. Apparently, $x^k_{k,j} = m_{k,j}, \forall k,j$ holds true in (4).

The significance of adopting MDS codes is to avoid content overlap among the fragments stored in different caches, hence reducing the average UA cost. Suppose that SBS k stores $m_{k,j}$ different fragments randomly drawn among the n_j fragments and $x^k_{t,j}$ of the $m_{k,j}$ fragments are randomly selected to be sent to SBS t. It is difficult to ensure that the fragments from the neighboring cells are always mutually exclusive. Thus, both the number of fragments stored in local cache and sent to other cells and which fragments being cached and shared contribute in deciding the backhaul rate and the average UA cost.

Lemma 1: Given any cooperative caching policy satisfying constraints (4b)–(4d), the UA cost in the coded scenario is always lower than the associated cost in the uncoded scenario defined as $C^{\text{uncoded}}_{\text{coop}}$, i.e., $C^{\text{MDS}}_{\text{coop}} \le C^{\text{uncoded}}_{\text{coop}}$.

Proof: See Appendix A. ■

We can tackle (4) by proving that the optimal cooperative caching policy always satisfies $\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j} \le 1, \forall k,j$. Letting $(\{\tilde{x}^t_{k,j}\},\{\tilde{m}_{k,j}\})$ be the optimal solution to (4) with at least a group of (k^*,j^*) satisfying $\sum_{t=1}^{K}\frac{\tilde{x}^t_{k^*,j^*}}{n_j} > 1$, we can always find some $(\{x^t_{k,j}\},\{\tilde{m}_{k,j}\})$ with $x^t_{k,j} = \tilde{x}^t_{k,j}, \forall (k,j,t) \ne (k^*,j^*,t)$ and $\sum_{t=1}^{K}\frac{\tilde{x}^t_{k^*,j^*}}{n_j} = 1$ which satisfy all the constraints in (4) while demanding the same cost from backhaul but a lower cost from content sharing among the cooperative SBSs. Consequently, the average UA cost is given by

$$C^{\text{MDS}}_{\text{coop}} = \sum_{j=1}^{N}\sum_{k=1}^{K}\left[\left(1-\sum_{t=1}^{K}z^t_{k,j}\right)f^M_k + \sum_{t=1}^{K}z^t_{k,j}f^t_k\right] \times s_j(1-\alpha_{k,j}), \quad (5)$$

where we let $q_{k,j} = \frac{m_{k,j}}{n_j}$ and $z^t_{k,j} = \frac{x^t_{k,j}}{n_j}$. Problem (4) can then be rewritten as

$$\min_{\{q_{k,j}\},\{z^t_{k,j}\}} \quad (5) \quad (6a)$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\sum_{j=1}^{N}q_{k,j}s_j \le M, \quad (6b)$$

$$0 \le q_{k,j} \le 1,\ \forall k,j, \quad (6c)$$

$$\sum_{t=1}^{K}z^t_{k,j} \le 1,\ \forall k,j, \quad (6d)$$

$$0 \le z^t_{k,j} \le q_{t,j},\ \forall k,j,t, \quad (6e)$$

which is linear and can easily be solved using, e.g., CVX.

For comparison, the average UA cost in the unicast based non-cooperative caching scenario is given by

$$C^{\text{unicast}}_{\text{noncoop}} = \sum_{j=1}^{N}\sum_{k=1}^{K}(1-q_{k,j})\,f^M_k s_j(1-\alpha_{k,j}). \quad (7)$$

As $f^t_k \le f^M_k$ and $z^k_{k,j} = q_{k,j}, \forall k,t,j$, we have

$$C^{\text{MDS}}_{\text{coop}} \le \sum_{j=1}^{N}\sum_{k=1}^{K}\left(1-\sum_{t=1}^{K}z^t_{k,j}+\sum_{t\ne k}z^t_{k,j}\right) \times f^M_k s_j(1-\alpha_{k,j}) \le C^{\text{unicast}}_{\text{noncoop}}. \quad (8)$$

IV. Multicast-Aware Cooperative Caching

In this section, a compound caching policy named multicast-aware cooperative caching is proposed. Taking the advantages of both multicasting at the MBS and collaboration among the SBSs, the UA cost of the network can be further reduced.

Unlike the literature where the knowledge of the actual requests from the cells was usually assumed, we analyze all possible request profiles and their probabilities using the learned file popularity. Here, the joint user request profile in all the cells is focused rather than the user request profiles in individual cells to capture the multicasting opportunities among the SBSs requiring the same file. In this case, the average backhaul load is defined as the average volume of the file packets requiring to be fetched from the MBS via backhaul with a single multicast transmission instead of multiple unicast transmissions to the SBSs requiring the same file in terms of all possible user request profiles.

We let Π_j be the collection of all the possible user request profiles and $\pi_j \in \Pi_j$ denote a particular user request profile for file j in all cells. Given any user request profile π_j, $\mathcal{K}_{\pi_j}$ is used to denote the set of the cells where file j is required by the served users. In case that file j is requested in all the cells except cell K, we have $\pi_j = [1, 1, \ldots, 1, 0]_{1\times K}$ where 1 means that file j is requested by users in the considered cell while 0 states that none of the users in the cell requests the file. Therefore, it follows that $\mathcal{K}_{\pi_j} = \{1, 2, \ldots, K-1\}$ for the mentioned π_j. The joint user request profile for all the files simultaneously can be written as $\{\pi_1, \ldots, \pi_N\}$. For each file j, if there are $t(\leq K)$ cells where the served users request file j, the corresponding file request profile π_j and the cell set $\mathcal{K}_{\pi_j}$ may have $\binom{K}{t}$ possible combinations.

In consideration of multicast-aware cooperative caching, the UA cost can be written as

$$C^{\text{MDS}}_{\text{mult,coop}} = \sum_{\{\pi_1,\ldots,\pi_N\}} \sum_{j=1}^{N} \Bigg[\left(1 - \min_{k\in\mathcal{K}_{\pi_j}} \sum_{t=1}^{K} z^t_{k,j}\right) \max_{k\in\mathcal{K}_{\pi_j}} f^M_k + \sum_{k\in\mathcal{K}_{\pi_j}} \sum_{t=1}^{K} z^t_{k,j} f^t_k \Bigg] s_j P_r(\{\pi_1,\ldots,\pi_N\}). \quad (9)$$

where $P_r(\{\pi_1, \ldots, \pi_N\})$ denotes the joint probability that a certain user request profile for all the files, i.e., $\{\pi_1, \ldots, \pi_N\}$ appears.

Lemma 2: Based on the fact that the backhaul load and shared content for a particular file j only relies on π_j regardless of $\{\pi_i\}_{i\neq j}$, the UA cost in (9) can be rewritten as

$$C^{\text{MDS}}_{\text{mult,coop}} = \sum_{j=1}^{N} \Bigg[\sum_{\pi_j\in\Pi_j} \left(1 - \min_{k\in\mathcal{K}_{\pi_j}} \sum_{t=1}^{K} z^t_{k,j}\right) \max_{k\in\mathcal{K}_{\pi_j}} f^M_k \times P_r(\pi_j) + \sum_{k=1}^{K} \sum_{t=1}^{K} z^t_{k,j} f^t_k (1-\alpha_{k,j}) \Bigg] s_j, \quad (10)$$

where $P_r(\pi_j)$ is the probability that π_j appears.

Proof: See Appendix B. ■

The average UA cost minimization problem is

$$\min_{\{q_{k,j}\},\{z^t_{k,j}\}} C^{\text{MDS}}_{\text{mult,coop}} \quad \text{s.t.} \quad \text{(6b)–(6e)}. \quad (11)$$

We recognize that similar content in different cells is preferred for multicast-aware caching while for cooperative caching the cached content in different cells should be mutually exclusive. The use of MDS codes strikes a balance in the combination.

Lemma 3: Given any mutlicast-aware cooperative caching policy $(\{q_{k,j}\}, \{z^t_{k,j}\})$ satisfying the constraints in (11), the UA cost in the coded scenario is always much lower than that in the uncoded case, i.e., $C^{\text{MDS}}_{\text{mult,coop}} \leq C^{\text{uncoded}}_{\text{mult,coop}}$.

Proof: See Appendix C. ■

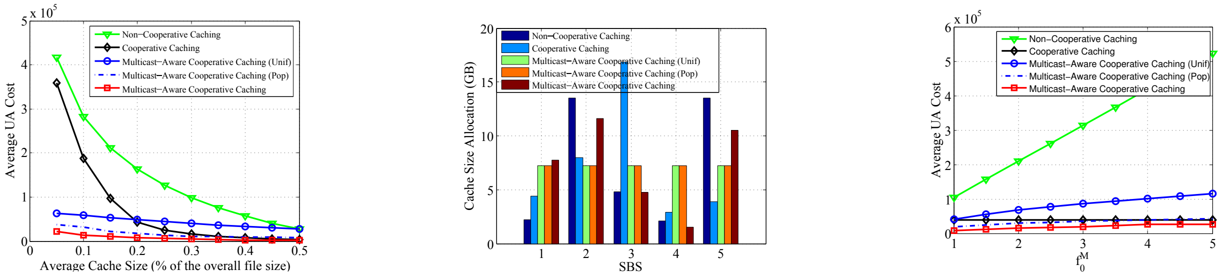
To solve (11), we resort to a greedy algorithm by listing all possible user request profiles for each file. Furthermore, a number of new variables and constraints needs to be added to linearize the function $\min(\cdot)$. That is, for any user request profile π_j, we introduce a new variable ξ_{π_j} subject to the constraints, i.e., $(0 \leq \xi_{\pi_j} \leq \sum_{t=1}^{K} z^t_{k,j}, \forall k \in \mathcal{K}_{\pi_j})$, to replace $\min_{k\in\mathcal{K}_{\pi_j}} \sum_{t=1}^{K} z^t_{k,j}$ in (10). Since (11) can be linearized, general solvers can be employed to solve it for small-scale networks. In practical scenarios with dozens of BSs and thousands of files where the greedy approach is not viable, in-cluster cooperative caching schemes can be used by decomposing the SBSs into a series of disjoint clusters and enabling the SBSs within the same cluster to cooperate to share cached content.

V. Simulation Results

In this section, simulation results are presented to compare the performance of the proposed cooperative caching schemes with that of the non-cooperative scheme in terms of the average UA cost. A small scale network with $K = 5$ SBSs is considered where the MBS is located at the center of the macro cell with radius $R = 400$km while the SBSs are randomly deployed uniformly within the cell without coverage overlapping. The overall cache capacity budget is presented as the average cache size for each SBS scaled by the overall file size given by $\rho = M/K/\sum_j s_j$. Unless otherwise specified, we set $\rho = 0.25, N = 100$ with the file sizes randomly chosen uniformly within $[0, 500]$MB. The skewness parameters $\{\gamma_k\}$ are selected randomly within $[0, 2]$ while the popularity ranks of the files in each cell are generated randomly. Also, the number of users in each cell is set to be ranged within $[0, 10]$, respectively. Here we assume that two SBSs can share content in their caches when the cost for retrieving content from the other SBS is lower than that of fetching content from the MBS. The unit cost coefficients for the two routes for fetching content from external storage are set as $f^M_0 = 2$ and $f_0 = 1$, unless otherwise specified.

Results for common cache management schemes such as the uniform content placement and popularity based content placement are provided and compared. For the two common schemes, we assume uniform cache size allocation among the SBSs, i.e., $M_k = M/K, \forall k, \forall j$. The former assumes that the cache capacity is equally allocated to the files while the later allocates the storage to most popular items in priority. Unless otherwise specified, unicast based content delivery and non-cooperative cache management are utilized.

As can be observed in Fig. 2, the proposed multicast-aware cooperative caching scheme shows the best performances followed by the unicast based cooperative caching scheme while the non-cooperative caching scheme yields the worst performance in all the cases. In addition, the multicast-aware cooperative caching schemes using common content placement demand higher UA costs compared with the proposed optimal multicast-aware caching scheme as expected.



(a) Impact of overall cache size M.

(b) Allocated cache sizes ($\rho = 0.25$).

(c) Impact of cost coefficient f_0^M.

Fig. 2: The average UA cost of the proposed cooperative caching schemes versus the non-cooperative scheme.

As we see in Fig. 2a, the UA costs decrease with the overall cache size in all cases. Apparently, the utility of cooperation in caching and multicast-aware caching reduce the average UA cost in the network dramatically.

Fig. 2b presents the cache size allocation among the SBSs using different caching schemes when $\rho = 0.25$. Results show that the optimal cache sizes for different cells are always heterogeneous as opposed to the assumption of uniform cache size allocation in many caching networks.

Finally, the impact of the ratio between the unit cost coefficients is studied in Fig. 2c where $f_0 = 1$ but f_0^M varies. Apparently, the UA cost of the non-cooperative caching scheme is proportional to f_0^M while the cooperative schemes have much better tolerance towards the increase of f_0^M for fetching content via backhaul.

VI. Conclusions

In this paper, we considered the design of content caching and sharing for cache-enabled heterogeneous small cell networks. We developed the cooperative caching schemes utilizing unicast and multicast content delivery between the MBS and the SBSs, respectively, for minimizing the long-term average UA cost subject to the overall cache capacity constraint. In both cases, we have obtained the optimal content placement by reformulating the original problems into convex ones with the analysis of the advantages of utilizing MDS codes, and cooperative caching over common caching schemes.

Appendix A

Given some cooperative caching policy $(\{x_{k,j}^t\}, \{m_{k,j}\})$, the costs for fetching content from neighboring cells are the same in the coded and uncoded caching scenarios. Therefore, the difference in the backhaul cost shows up most clearly in the UA costs. When uncoded fragments are stored, all the fragments except the ones that are either stored in local cache or fetched from the neighboring cells are needed from the MBS via backhual to each cell requesting the particular file. Considering the possible content overlap amongst those fragments, the number of unique fragments for file j available at cell $k \in \mathcal{K}_{\pi_j}$ would always be less than or equal to $\sum_t x_{k,j}^t$ for a certain user request profile π_j which leads to a higher backhaul rate than that in the MDS coded case. If the fragments are assumed to be randomly selected to be stored in the cells and then sent to the neighboring cells *equiprobably*, the probability of each fragment of file j needing to be sent to cell k via backhaul, i.e., not being stored locally or sent to the particular cell k from other SBSs, would be given by

$$\widehat{\rho}_{k,j} = \prod_{t=1}^{K} \left(\frac{\binom{n_j - 1}{m_{t,j}}}{\binom{n_j}{m_{t,j}}} + \frac{\binom{n_j - 1}{m_{t,j} - 1}}{\binom{n_j}{m_{t,j}}} \frac{\binom{m_{t,j} - 1}{x_{k,j}^t}}{\binom{m_{t,j}}{x_{k,j}^t}} \right) = \prod_{t=1}^{K} \left(1 - \frac{x_{k,j}^t}{n_j} \right). \tag{12}$$

In this case, the average UA cost can be written as

$$C_{\text{coop}}^{\text{uncoded}} = \sum_{j=1}^{N} \sum_{k=1}^{K} \left[\widehat{\rho}_{k,j} f_k^M + \sum_{t=1}^{K} \frac{x_{k,j}^t}{n_j} f_k^t \right] s_j (1 - \alpha_{k,j}). \tag{13}$$

Compared with the UA cost in (8), if we can prove that

$$\prod_{t=1}^{K} \left(1 - \frac{x_{k,j}^t}{n_j} \right) \geq 1 - \min\left(1, \sum_{t=1}^{K} \frac{x_{k,j}^t}{n_j} \right), \forall k, j, \tag{14}$$

then it holds true that $C_{\text{coop}}^{\text{MDS}} \leq C_{\text{coop}}^{\text{uncoded}}$. Hence, here we focus on the proof of the result (14). As can be observed, when $\sum_{t=1}^{K} \frac{x_{k,j}^t}{n_j} \geq 1$, (14) is always true. When $\sum_{t=1}^{K} \frac{x_{k,j}^t}{n_j} < 1$, the right hand side of (14) equals to $\left(1 - \sum_{t=1}^{K} \frac{x_{k,j}^t}{n_j} \right)$. In this case, we prove (14) using mathematical induction.

To be brief, we mathematically reformulate the problem into a general problem, which reads

$$\prod_{t=1}^{K} (1 - \chi_t) \geq 1 - \sum_{t=1}^{K} \chi_t, \tag{15}$$

where $\chi_t \in [0, 1]$. Obviously, when $K = 1$ or 2, the statement is always true as expected. Now assuming that (15) holds for $K = \kappa$, we hence have

$$\prod_{t=1}^{\kappa} (1 - \chi_t) \geq 1 - \sum_{t=1}^{\kappa} \chi_t. \tag{16}$$

Then it follows that

$$\begin{aligned} \prod_{t=1}^{\kappa+1} (1 - \chi_t) &= \prod_{t=1}^{\kappa} (1 - \chi_t) - \prod_{t=1}^{\kappa} (1 - \chi_t) \chi_{\kappa+1} \\ &\geq \left(1 - \sum_{t=1}^{\kappa} \chi_t \right) - \chi_{\kappa+1}, \end{aligned} \tag{17}$$

due to the fact that $0 \leq \prod_{t=1}^{\kappa} (1 - \chi_t) \leq 1$ as well as the inequality (16). Now we are able to conclude that the statement is true for all available K via induction. Then going back to the original problem and letting $\chi_t = \frac{x^t_{k,j}}{n_j}$ for any given k, we have proved the statement

$$\prod_{t=1}^{K}\left(1-\frac{x^t_{k,j}}{n_j}\right) \geq 1-\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}, \forall k, j. \tag{18}$$

Based on this analysis, $C^{\text{MDS}}_{\text{coop}} \leq C^{\text{uncoded}}_{\text{coop}}$ is then proved.

Appendix B

In (9), the instantaneous UA costs for all kinds of possible user request profiles $\{\pi_1, \ldots, \pi_N\}$ are summed up to obtain the average UA cost while that for a particular user request profile is composed of the associated UA costs for all the files. Equivalently, the average backhaul rate can also be calculated by summing up the average UA cost for each file in terms of all kinds of possible user request profiles. Mathematically, we are able to rewrite (9) as

$$C^{\text{MDS}}_{\text{mult,coop}} = \sum_{j=1}^{N}\sum_{\{\pi_1,\ldots,\pi_N\}}\left[\left(1-\min_{k\in\mathcal{K}_{\pi_j}}\sum_{t=1}^{K} z^t_{k,j}\right)\max_{k\in\mathcal{K}_{\pi_j}} f^M_k + \sum_{k\in\mathcal{K}_{\pi_j}}\sum_{t=1}^{K} z^t_{k,j} f^t_k\right] s_j P_r(\{\pi_1,\ldots,\pi_N\}). \tag{19}$$

For a particular file j, the UA cost is subject to the shared content $z^t_{k,j}$ and the associated user request profile π_j regardless of the profiles for other files $\{\pi_i\}_{i\neq j}$. That is to say, any user request profile $\{\pi_1, \ldots, \pi_N\}$ with the same π_j would yield the same UA cost for file j. Consequently, when calculating the UA cost for a file, we can only consider different user request profiles for the certain file and ignore the user request profiles for other files. Hence, (9) can be further reformulated into

$$C^{\text{MDS}}_{\text{mult,coop}} = \sum_{j=1}^{N}\sum_{\pi_j\in\Pi_j}\left[\left(1-\min_{k\in\mathcal{K}_{\pi_j}}\sum_{t=1}^{K} z^t_{k,j}\right)\max_{k\in\mathcal{K}_{\pi_j}} f^M_k + \sum_{k\in\mathcal{K}_{\pi_j}}\sum_{t=1}^{K} z^t_{k,j} f^t_k\right] P_r(\pi_j) s_j. \tag{20}$$

As we can see, the first item denotes the backhaul cost while the second item presents the cost for content sharing among the cooperative SBSs. For each given user request profile for a particular file π_j, the cost for fetching content from the cooperative SBSs at cell k appears only when file j is requested by the users in cell k which means that $\pi_j(k) = 1$ regardless of the individual user request profiles in other cells. It is easy to prove $P_r(\pi_j|_{\pi_j(k)=1}) = 1 - \alpha_{k,j}$, and so (10).

Appendix C

If $(\{x^t_{k,j}\}, \{m_{k,j}\})$ is given, then the costs for fetching content from neighboring cells will be the same in the coded and uncoded caching scenarios. As a result, the comparison is focused on the backhaul costs in the two scenarios. When uncoded fragments are stored, all the fragments except for the ones that can be fetched at all of the cells requesting the file either from local cache or from the neighboring cells are needed to be sent from the MBS via multicast transmission. Assuming that the fragments are randomly selected to be stored in the cells and then sent to the neighboring cells *equiprobably*, the probability of each fragment of file j available at all of the cells requesting the file either from local cache or from the neighboring cells would be given by

$$\widetilde{\rho}_{\pi_j} = \prod_{k\in\mathcal{K}_{\pi_j}} (1-\widehat{\rho}_{k,j}), \tag{21}$$

where $\widehat{\rho}_{k,j}$ is the probability of each fragment of file j not being stored locally or sent to the particular cell k from other SBSs given by (12) in Appendix A. Similar to the multicast-aware case, the average UA cost can be written as

$$C^{\text{uncoded}}_{\text{mult,coop}} = \sum_{j=1}^{N}\left[\sum_{\pi_j\in\Pi_j}\left(1-\widetilde{\rho}_{\pi_j}\right)\max_{k\in\mathcal{K}_{\pi_j}} f^M_k P_r(\pi_j) + \sum_{k=1}^{K}\sum_{t=1}^{K} z^t_{k,j} f^t_k (1-\alpha_{k,j})\right] s_j. \tag{22}$$

According to (12) and (18), we obtain

$$\widetilde{\rho}_{\pi_j} \leq \prod_{k\in\mathcal{K}_{\pi_j}} (\sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j}). \tag{23}$$

As $0 \leq \sum_{t=1}^{K}\frac{x^t_{k,j}}{n_j} \leq 1, \forall k \in \mathcal{K}_{\pi_j}$, it holds true that $\widetilde{\rho}_{\pi_j} \leq \min_{k\in\mathcal{K}_{\pi_j}}\sum_{t=1}^{K} z^t_{k,j}$. Compared with the average UA cost in (10), we derive that $C^{\text{MDS}}_{\text{mult,coop}} \leq C^{\text{uncoded}}_{\text{mult,coop}}$.

References

[1] G. Paschos, E. Bastug, I. Land, G. Caire and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16–22, Aug. 2016.

[2] V. Bioglio, F. Gabry, and I. Land, "Optimizing MDS codes for caching at the edge," in *Proc. IEEE GLOBECOM*, San Diego, USA, Dec. 2015.

[3] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2995–3007, Jan. 2016.

[4] J. Liao, K. K. Wong, M. R. A. Khandaker, and Z. Zheng, "Optimizing cache placement for heterogeneous small cell networks," *IEEE Commun. Letters*, no.99, pp.1–1, Sept. 2016.

[5] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Trans. Mobile Computing*, vol. 15, no. 10, pp. 1863–1876, Aug. 2016.

[6] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "In-network caching and content placement in cooperative small cell networks," in *Proc. 1st Int. Conf. 5G for Ubiquitous Connectivity*, pp.128–133, Nov. 2014.

[7] X. Peng, J. Zhang, S. H. Song, and K. B. Letaief, "Cache size allocation in backhaul limited wireless networks," in *Proc. IEEE Int. Conf. Commun.(ICC)*, pp. 1–6, May 2016.

[8] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Trans. Computers*, vol. 65, no. 1, pp. 95–107, Jan. 2016.

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

[10] D. Lopez-Perez, I. Guvenc, G. Roche, M. Kountouris, T. Quek, and J. Zhang, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 22–30, Jun. 2011.

[11] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. of IEEE INFOCOM 99*, New York, U.S.A, Mar. 1999.

[12] M. Grant, and S. Boyd, *CVX: Matlab software for disciplined convex programming*, version 2.0 beta. http://cvxr.com/cvx, Sept. 2013.