Optimizing Cache Placement for Heterogeneous Small Cell Networks

Jialing Liao, Kai-Kit Wong, *Fellow, IEEE*, Muhammad R. A. Khandaker, *Member, IEEE*, and Zhongbin Zheng

***Abstract*—In this letter, we study the optimization for cache content placement to minimize the backhaul load subject to cache capacity constraints for caching enabled small cell networks with heterogeneous file and cache sizes. Multicast content delivery is adopted to reduce the backhaul rate exploiting the independence among maximum distance separable coded packets.**

***Index Terms*—Heterogeneous networks, cache storage.**

I. Introduction

TO ACHIEVE the targets of the fifth-generation (5G) cellular communication systems, an effective solution is to cache popular files at the network edge before users request them. In [1], the authors first viewed caching from the perspective of information theory and the work was later extended to general scenarios with nonuniform cache sizes, file sizes and file popularity in [2]. In [3] and [4], beamforming design was studied to minimize the backhaul cost and transmit power. In addition, learning, matching and online algorithms were used to solve physical-layer caching problems in [5]–[7].

Physical-layer caching is thought to be of great importance to small cell networks where popular contents are brought to the local servers right at the small-cell base stations (SBSs). Similar to [8] and [9], this letter studies the optimal cache content placement for caching enabled small cell networks. In [8], coded caching using unicast was studied while [9] was focused on uncoded caching using multicast in delay tolerant networks.

This letter aims to obtain the optimal (offline) cache content placement for minimizing the backhaul rate subject to cache capacity constraints for small cell networks in which maximum distance separable (MDS) codes are adopted.[1] Unlike [8] considering an unlikely setting of identical content placement in all caches with homogeneous settings, we consider a more practical scenario with heterogeneous file and cache sizes, and in this case the content placement in different caches will not always be the same and hence it is no longer available for the macro base station (MBS) to deliver the uncached content via a shared link. To tackle this problem, we utilize the independence among MDS coded packets and a near-optimal solution is obtained using a specific solver for mixed integer linear program (MILP) after a series of reformulations.

Manuscript received July 29, 2016; revised September 12, 2016; accepted September 19, 2016. Date of publication September 21, 2016; date of current version January 6, 2017. This work is supported by EPSRC under grant EP/K015893/1. The associate editor coordinating the review of this letter and approving it for publication was J. Lee.

J. Liao, K.-K. Wong, and M. R. A. Khandaker are with the Department of Electronic and Electrical Engineering, University College London, WC1E 7JE, U.K. (e-mail: jialing.liao.14@ucl.ac.uk; kai-kit.wong@ucl.ac.uk; m.khandaker@ucl.ac.uk).

Z. Zheng is with the East China Institute of Telecommunications, China Academy of Information and Communications Technology, 200001, China (e-mail:ben@ecit.org.cn)

Digital Object Identifier 10.1109/LCOMM.2016.2612197

[1]Note that the algorithm in [9] is only applicable to the uncoded caching case, so will be unusable for our problem as MDS codes are used.

II. System Model

We consider a small cell network comprising a single MBS, and K small cells each consisting of a single SBS and I_k users among which each SBS can only answer to the requests of a maximum of $I(I \geq I_k, \forall k)$ users at the same time. The requests of the remaining users are served by the MBS. It is assumed that there is no coverage overlapping amongst all the SBSs which operate in sub-channels disjoint with the MBS. Moreover, enhanced inter-cell interference coordination techniques (eICIC) or/and orthogonal spectra are utilized by the neighboring SBSs [10], [11]. We also assume that the MBS has access to all the files defined as $\mathcal{F} \triangleq \{f_1, f_2, \ldots, f_N\}$ with distinct file sizes $\mathbf{s} = [s_1, s_2, \ldots, s_N]$. The users located outside of the small cells can only be served by the MBS and hence are ignored when considering the backhaul rate from the MBS to the SBSs. Note that it is also assumed that each user is able to request one file at one time slot. Instead of assuming identical cache size in all SBSs which is difficult to satisfy in practice, here we consider that the SBSs have heterogeneous cache sizes. We let $M_k(M_k \leq \sum_{j=1}^{N} s_j)$ be the cache size in SBS k.

Using MDS codes parametrized by (l_j, n_j), file j is equally cut into n_j fragments and coded into l_j independent packets, any n_j of which can recover the file. The SBS in cell k caches $m_{k,j}$ coded packets of file j. Let $\mathbf{m}^j = [m_{1,j}, \ldots, m_{K,j}]$. The SBSs push the cached packets to the users when requested while the uncached parts are delivered to the SBSs via the backhaul from the MBS. Taking advantages of the independence among the MDS codes, we adopt *multicasting* between MBS and SBSs to reduce the backhaul rate.

To recover the requested file with minimum redundancy, file j is coded into $l_j = \sum_{k=1}^{K} m_{k,j} + n_j - \min_{k=1}^{K} m_{k,j}$ packets to ensure that the uncached packets delivered from the MBS are different from all the cached packets, even in an extreme case that the SBSs store totally different packets.

III. Content Placement Optimization

Let $\pi_j \in \Pi_j$ denote the user request profile for file j in all cells in which Π_j is the collection of all the possible user request profiles for file j. Given the user request profile π_j, we use $\mathcal{K}_{\pi_j}$ to denote the collection of the cells where file j is required. For instance, if π_j demonstrates that file j is requested by all the cells except cell K, i.e., $\pi_j = [1, 1, \ldots, 1, 0]$, then we will have $\mathcal{K}_{\pi_j} = \{1, 2, \ldots, K-1\}$. In addition, if there are $t(\leq K)$ cells requiring file j, π_j and the corresponding $\mathcal{K}_{\pi_j}$ may have $\binom{K}{t}$ possible combinations. The total number of different π_j and $\mathcal{K}_{\pi_j}$ will be as high as 2^K.

Our aim is to minimize the average backhaul load, i.e., the volume of the file packets needed to be delivered via backhaul using multicasting, subject to the cache capacity constraints by optimizing the cache content placement. The average backhaul rate is obtained by taking expectation of the instantaneous backhaul rate with respect to the joint probability of user request profile for all the files $\{\pi_1, \ldots, \pi_N\}$. That is,

$$\min_{\{m_{k,j}\}} \sum_{\{\pi_1,\ldots,\pi_N\}} \sum_{j=1}^{N} \left(1 - \min_{k\in\mathcal{K}_{\pi_j}} \frac{m_{k,j}}{n_j}\right) s_j P_r(\{\pi_1, \ldots, \pi_N\}) \quad (1a)$$

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

$$0 \le m_{k,j} \le n_j, \quad \forall k, j, \quad (1c)$$

where $P_r(\{\pi_1, \ldots, \pi_N\})$ shows the joint probability that a certain user request profile, i.e., $\{\pi_1, \ldots, \pi_N\}$ appears, and s_j denotes the size for file j.

Lemma 1: Based on the fact that the backhaul load for a particular file j only relies on π_j regardless of $\{\pi_i\}_{i\neq j}$, the average backhaul rate in (1a) can be rewritten as

$$C^{\text{MDS}}_{\text{multicast}} = \sum_{j=1}^{N} \sum_{\pi_j\in\Pi_j} \left(1 - \min_{k\in\mathcal{K}_{\pi_j}} \frac{m_{k,j}}{n_j}\right) s_j P_r(\pi_j). \quad (2)$$

where $P_r(\pi_j)$ is the probability a certain profile π_j appears.

Proof: See Appendix A. ■

To show the advantages of storing MDS coded packets over storing the uncoded segments directly in our settings, we assume that the SBS in cell k stores m_j^k different fragments randomly drawn among the n_j fragments. Let $\mathcal{M}_j$ show the detail of the fragments of file j stored in the caches and $d\left(\mathcal{M}_j, \mathcal{K}_{\pi_j}\right)$ denote the number of same fragments stored in all the cells requesting file j. The backhaul rate is given by

$$C^{\text{Uncoded}}_{\text{multicast}} = \sum_{j=1}^{N} \sum_{\pi_j\in\Pi_j} \left(1 - \frac{d\left(\mathcal{M}_j, \mathcal{K}_{\pi_j}\right)}{n_j}\right) s_j P_r(\pi_j). \quad (3)$$

Due to the fact that the number of same fragments stored in all the cells requesting file j is always less than or equal to the minimum number of the fragments stored in those cells, i.e., $d\left(\mathcal{M}_j, \mathcal{K}_{\pi_j}\right) \le \min_{k\in\mathcal{K}_{\pi_j}} m_{k,j}$, the use of MDS codes clearly helps reduce the average backhaul rate. If the uncoded segments are assumed to be randomly drawn among the n_j fragments *equiprobably*, the probability of each fragment of file j being stored in all the cells requesting the file will be

$$\rho_j = \prod_{k\in\mathcal{K}_{\pi_j}} \frac{\binom{n_j-1}{m_{k,j}-1}}{\binom{n_j}{m_{k,j}}} = \prod_{k\in\mathcal{K}_{\pi_j}} \frac{m_{k,j}}{n_j}. \quad (4)$$

As $\frac{m_{k,j}}{n_j} \le 1, \forall k \in \mathcal{K}_{\pi_j}$, it holds true that $\rho_j \le \min_{k\in\mathcal{K}_{\pi_j}} \frac{m_{k,j}}{n_j}$. In this case, the expectation of d in terms of different $\mathcal{M}_j$ with given $\mathbf{m}^j$ and π_j is given by $\overline{d}\left(\mathbf{m}^j, \mathcal{K}_{\pi_j}\right) = n_j\rho_j \le \min_{k\in\mathcal{K}_{\pi_j}} m_{k,j}$. The same conclusion can be drawn.

Now assuming that the popularity of the files is arranged in a descending order according to the Zipf's law, the frequency for file j to be requested by each user can be written as [12]

$$p_j = \frac{(1/j^\gamma)}{\sum_{i=1}^{N}(1/i^\gamma)}, \ \forall j, \quad (5)$$

where $\gamma \in [0.5, 1.5]$ is the skewness reflecting the concentration of the popularity distribution. Hence, the probability of file j not being requested by the users in the cell is

$$\alpha_j = (1 - p_j)^I, \quad \forall j. \quad (6)$$

Though $P_r(\pi_j)$ can be calculated by (6), it would be difficult to fully list all possible user request profiles and analyze the objective function. However, if we know the relationships amongall the elements in $\mathbf{m}^j$, a closed-form expression of the objective function can be obtained in the following lemma.

Lemma 2: Let $r_{k,j}$ denote the rank of the value of $m_{k,j}$ in $\mathbf{m}^j$. For instance, $r_{k,j} = 1$ means $m_{k,j}$ is the smallest while $r_{k,j} = K$ states that $m_{k,j}$ is the largest in $\mathbf{m}^j$. The objective function (1a) can then be rewritten as

$$\widetilde{C}^{\text{MDS}}_{\text{multicast}} = \sum_{j=1}^{N} \sum_{k=1}^{K} \left(1 - \frac{m_{k,j}}{n_j}\right) s_j (1 - \alpha_j)\alpha_j^{(r_{k,j}-1)}. \quad (7)$$

Proof: Firstly, we divide the possible user request profiles for each file, e.g., π_j into $K+1$ types defined as $\{\pi_j^0, \pi_j^1, \pi_j^2, \ldots, \pi_j^K\}$ according to the different values of the associated backhaul load for file j, i.e., $\{0, 1-\frac{m_{1,j}}{n_j}, 1-\frac{m_{2,j}}{n_j}, \ldots, 1-\frac{m_{K,j}}{n_j}\}$, respectively. Note that when file j is not requested by any of the cells, the backhaul is not needed. If cell k stores the least number of packets of file j among all the cells requesting file j, i.e., $\min_{t\in\mathcal{K}_{\pi_j}} \frac{m_{t,j}}{n_j} = \frac{m_{k,j}}{n_j}$, then the associated user request profile π_j^k will imply that file j is requested by cell k and that probably some cells have cached more packets of file j but there will not be any cell t satisfying $r_{t,j} < r_{k,j}$, i.e., $m_{t,j} \le m_{k,j}$. Hence, we have $P_r(\pi_j^k) = (1-\alpha_j)\alpha_j^{(r_{k,j}-1)}$. Finally, after summing up all types of user request profiles $\{\pi_j^k\}$ for all files, the average backhaul rate can be written as (7). ■

As a comparison, in the typical unicast case without coverage overlap among the SBSs, the backhaul rates for storing uncoded fragments directly or coded packets would be [8]

$$C_{\text{unicast}} = \sum_{j=1}^{N} \sum_{k=1}^{K} \left(1 - \frac{m_{k,j}}{n_j}\right) s_j (1 - \alpha_j). \quad (8)$$

Note that after using multicast, additional multipliers $0 < \alpha_j^{(r_{k,j}-1)} \le 1, \forall k, \forall j$ appear and hence bring a global gain.

Substituting (7) into (1), the problem of interest becomes

$$\min_{\{m_{k,j}\}} \ \widetilde{C}^{\text{MDS}}_{\text{multicast}} \text{s.t. (1b) and (1c).} \quad (9)$$

Note that we can separate the files into an arbitrary number of fragments. We define $q_{k,j} \triangleq \frac{m_{k,j}}{n_j}$ as the cached percentage of file j in SBS k. Accordingly, we let $\mathbf{q}^j = [q_{1,j}, q_{2,j}, \ldots, q_{K,j}]$ and their ranks remain the same. Then (9) is rewritten as

$$\min_{\{q_{k,j}\}} \ \sum_{j=1}^{N} \sum_{k=1}^{K} \left(1 - q_{k,j}\right) s_j (1 - \alpha_j)\alpha_j^{(r_{k,j}-1)} \quad (10a)$$

$$\text{s.t. } \sum_{j=1}^{N} q_{k,j} s_j \le M_k, \quad \forall k, \text{ and } 0 \le q_{k,j} \le 1, \quad \forall k, j. \quad (10b)$$

In (10), $\{q_{k,j}\}$ are to be optimized and hence unknown before the problem is solved. It is impossible to predict the ranks $\{r_{k,j}\}$ which depend on the values of $\{q_{k,j}\}$. To tackle this problem, we firstly sort $\mathbf{q}^j, \forall j$ in an ascending order and define the sorted variables as $\mathbf{g}^j = [g_{1,j}, g_{2,j}, \ldots, g_{K,j}]$ with $r_{k,j} = k$ in $\mathbf{g}^j, \forall j$. Problem (10) is then expressed as

$$\min_{\{q_{k,j}\},\{g_{k,j}\}} \sum_{j=1}^{N}\sum_{k=1}^{K} \left(1 - g_{k,j}\right) s_j (1-\alpha_j)\alpha_j^{(k-1)} \tag{11a}$$

$$\text{s.t. } \mathbf{g}^j = \text{sort}(\mathbf{q}^j), \quad \forall j, \text{ and (10b).} \tag{11b}$$

Nevertheless, sorting the variables to be optimized is definitely unconvex. The challenge then becomes how to demonstrate the relationships between $\mathbf{q}^j$ and $\mathbf{g}^j$ so as to satisfy the constraints.

Lemma 3: By introducing $\mathbf{X} = [x_{t,j}^k]_{K\times N\times K}$ with $x_{t,j}^k \in \{0,1\}$ such that $q_{k,j} = \sum_{t=1}^{K} g_{t,j}x_{t,j}^k$, we rewrite (11) as

$$\min_{\{g_{k,j}\},\{x_{t,j}^k\}} \sum_{j=1}^{N}\sum_{k=1}^{K} \left(1 - g_{k,j}\right) s_j (1-\alpha_j)\alpha_j^{(k-1)} \tag{12a}$$

$$\text{s.t. } \sum_{j=1}^{N}\sum_{t=1}^{K} g_{t,j}x_{t,j}^k s_j \le M_k, \quad \forall k, \tag{12b}$$

$$\sum_{t=1}^{K} x_{t,j}^k = 1, \quad \forall k, j, \tag{12c}$$

$$\sum_{k=1}^{K} x_{t,j}^k = 1, \quad \forall t, j, \tag{12d}$$

$$0 \le g_{k,j} \le 1, \quad \forall k, j, \tag{12e}$$

$$g_{k,j} \le g_{k+1,j}, \quad \forall k < K, \text{ and } \forall j, \tag{12f}$$

$$x_{t,j}^k \in \{0,1\}, \quad \forall t, j, k. \tag{12g}$$

Proof: See Appendix B. ■

Clearly, (12) is a mixed integer nonlinear program (MINLP) and hence cannot be solved directly. Therefore, we resort to linearizing the products of the variables in (12b).

Lemma 4: Let z be the product of a binary x and a continuous variable $y (0 \le y \le \tilde{y})$. We can linearize the equation $z = xy$ by adding the following constraints *equivalently*

$$z \le x\tilde{y} \text{ (a)}, \;\; z \ge y-(1-x)\tilde{y} \text{ (b)}, \;\; z \le y \text{ (c)}, \;\; z \ge 0 \text{ (d)}. \tag{13}$$

Proof: See Appendix C. ■

According to *Lemma 4*, we can easily replace the products in constraint (12b) with a new group of variables defined as $\mathbf{Z} = [z_{t,j}^k]_{K\times N\times K}$. Then (12) can be rewritten as

$$\min_{\{g_{k,j}\},\mathbf{X},\mathbf{Z}} \sum_{j=1}^{N}\sum_{k=1}^{K} \left(1 - g_{k,j}\right) s_j (1-\alpha_j)\alpha_j^{(k-1)} \tag{14a}$$

$$\text{s.t. } \sum_{j=1}^{N}\sum_{t=1}^{K} z_{t,j}^k s_j \le M_k, \quad \forall k, \tag{14b}$$

$$z_{t,j}^k \le x_{t,j}^k, \quad \forall t, j, k, \tag{14c}$$

$$z_{t,j}^k \ge g_{k,j} - (1 - x_{t,j}^k), \quad \forall t, j, k, \tag{14d}$$

$$z_{t,j}^k \le g_{k,j}, \quad \forall t, j, k, \tag{14e}$$

$$z_{t,j}^k \ge 0, \quad \forall t, j, k, \tag{14f}$$

$$\text{(12c)–(12g).} \tag{14g}$$

Based on the equivalence between the obtained linear constraints and the assumption of $z_{t,j}^k = g_{t,j}x_{t,j}^k$ with the basic settings of $\{g_{t,j}\}$ and $\{x_{t,j}^k\}$, it is apparent that the obtained solution to problem (14) will also be the solution to (12).

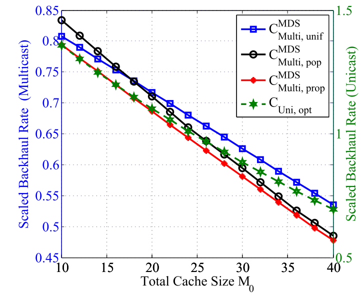

Fig. 1. The backhaul rates versus the total cache size M_0.

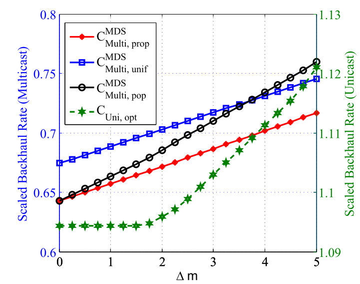

Fig. 2. The backhaul rates versus Δm.

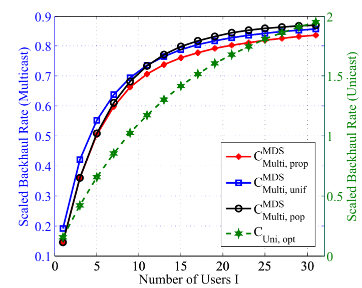

Fig. 3. The backhaul rates versus I.

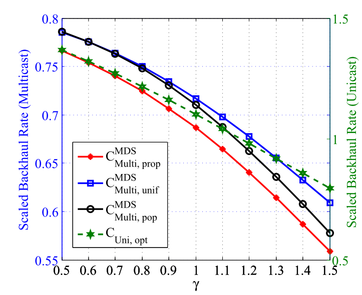

Fig. 4. The backhaul rates versus skewness γ.

IV. Simulation Results

Simulation results are now provided with comparison to the uniform and popularity based content placement schemes in [8] and [9]. To show the benefit of multicast, the unicast scenario in (8) is studied as well. We consider a small cell network with $K = 3$ cells and the cache sizes $M = [\frac{M_0}{3} + \Delta m, \frac{M_0}{3}, \frac{M_0}{3} - \Delta m]$ where M_0 is the total cache size while Δm is the cache size differentiation. We set $N = 10$ with file sizes randomly chosen uniformly between 1 and 5 independently. Unless otherwise specified, we set $M_0 = 20$, $\Delta\ m = 3$, $I = 10$, $\gamma = 1$. Note that the backhaul rates have been scaled with the total file size. The results in the multicast and unicast scenarios are presented using the left and right axes, respectively.

Fig. 1 studies the average backhaul rates for various total cache sizes. As we can observe, the backhaul rates decrease with the total cache size in all cases. Also, as expected, the multicast based schemes outperform the unicast scheme and the proposed scheme reduces the backhaul load among all.

Fig. 2 explores the impact of the cache size differentiation Δm on the backhaul rates with fixed M_0. Similar to Fig. 1, the proposed scheme shows the best performance. In addition, the backhaul rate reduction of the proposed scheme over the popularity based scheme increases drastically when improving Δm which illustrates the significance of the proposed scheme in small cell networks with heterogeneous cache sizes.

Fig. 3 illustrates the impact of the number of served users I in each cell on the backhaul rate, while the impact of the skewness γ is investigated in Fig. 4. Again, the multicast based schemes show better performance than the unicast based scheme and the proposed scheme yields the lowest backhaul rate. Moreover, the performance gain of the proposed scheme to the popularity based scheme changes little when increasing γ while that to the uniform scheme rises more obviously.

V. Conclusions

In this letter, the optimization of cache content placement was investigated for MDS coded caching enabled small cell networks with heterogeneous file and cache sizes. To minimize the average backhaul rate, multicast transmission was adopted. Results showed that the proposed scheme using MILP outperforms the existing schemes in terms of backhaul requirements.

Appendix A

In (1a), the instantaneous backhaul rates for all kinds of possible user request profiles $\{\pi_1, \ldots, \pi_N\}$ are summed up to obtain the average backhaul rate while that for a particular user request profile is composed of the associated backhaul rates for all the files. Equivalently, the average backhaul rate can also be calculated by summing up the average backhaul rate for each file in terms of all kinds of possible user request profiles. Mathematically, we are able to rewrite (1a) as

$$C^{\mathrm{MDS}}_{\mathrm{multicast}} = \sum_{j=1}^{N} \sum_{\{\pi_1,\ldots,\pi_N\}} \left(1 - \min_{k\in\mathcal{K}_{\pi_j}} \frac{m_{k,j}}{n_j}\right) s_j P_r(\{\pi_1, \ldots, \pi_N\}). \tag{15}$$

For a particular file j, the volume of packets to be sent via backhaul is subject to the content placement $\mathbf{m}^j$ 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 backhaul rate for file j. Consequently, when calculating the backhaul rate 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, (15) can be further reformulated into

$$C^{\mathrm{MDS}}_{\mathrm{multicast}} = \sum_{j=1}^{N} \sum_{\pi_j} \left(1 - \min_{k\in\mathcal{K}_{\pi_j}} \frac{m_{k,j}}{n_j}\right) s_j P_r(\pi_j), \tag{16}$$

which is the same as (2) found by considering the user profile for each file and then summing up the backhaul rate for all the files. The equivalence between (1a) and (2) is hence proved.

Appendix B

Since the solution of (11) always satisfies $\widetilde{\mathbf{g}}^j = \mathrm{sort}(\widetilde{\mathbf{q}}^j), \forall j$, it can be easily proved that $\widetilde{q}_{k,j} = \widetilde{g}_{\widetilde{r}_{k,j},j}$ where $\widetilde{r}_{k,j}$ is the rank of $\widetilde{q}_{k,j}$ in $\widetilde{\mathbf{q}}^j$. Note that the ranks must be unique integers. Hence, if we let $\widetilde{x}^k_{t,j} = 1|_{\widetilde{r}_{k,j}=t}$ and otherwise $\widetilde{x}^k_{t,j} = 0$, we will then get $\widetilde{q}_{k,j} = \sum_{t=1}^{K} \widetilde{g}_{t,j}\widetilde{x}^k_{t,j}$, which satisfy all the constraints in (12). Hence, $\{\widetilde{g}_{k,j}\}$ and $\{\widetilde{x}^k_{t,j}\}$ are the solution of (12). Oppositely, if $\{\widetilde{g}_{k,j}\}$ and $\{\widetilde{x}^k_{t,j}\}$ are known to be the solution to (12), it is easy to prove that $\{\widetilde{g}_{k,j}\}$ are the solution to (11) and then use them to recover $\{\widetilde{q}_{k,j}\}$, i.e., $\widetilde{q}_{k,j} = \sum_{t=1}^{K} \widetilde{g}_{t,j}\widetilde{x}^k_{t,j}$. In this case, the rank of $\widetilde{q}_{k,j}$ in $\widetilde{\mathbf{q}}^j$ is $\widetilde{r}_{k,j} = t|_{\widetilde{x}^k_{t,j}=1}$. The equivalence is therefore proved.

Appendix C

Firstly, we prove that any (x, y, z) with $z = xy$ can satisfy constraints (13a)–(13d). Based on the definition that $0 \le y \le \widetilde{y}$ and $x \in \{0, 1\}$, z is monotonically increasing with both x and y. Thus, (13a), (13c) and (13d) always hold. Also, when $x = 0$, we get $y - \widetilde{y} \le 0$ and $z = 0$ in (13b). Similarly, when $x = 1$, we can prove (13b). Now suppose that (x, y, z) satisfies (13a)–(13d) and we prove that $z = xy$ by contradiction. Assume that there is a z satisfying $z > xy$. According to (13c), we then get $z \le y$ which indicates that $x = 0$ and hence $z > 0$. This contradicts with (13a). The assumption cannot be true. Similarly, we can prove that once $z < xy$ happens, x must be equal to 1 and $z < y$ in order to satisfy (13d). This in turn violates (13b) which requires $z \ge y$. Consequently, $z = xy$ always holds in this case. *Lemma 4* is then proved.

References

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

[2] M. Ji *et al.*, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[3] M. Tao *et al.*, (Dec. 2015). "Content-centric sparse multicast beamforming for cache-enabled cloud RAN." [Online]. Avaliable: https://arxiv.org/abs/1512.06938

[4] A. Liu and V. K. N. Lau, "Mixed-timescale precoding and cache control in cached MIMO interference network," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6320–6332, Dec. 2013.

[5] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Commun. Conf. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 1897–1903.

[6] K. Hamidouche, *et al.* "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *Proc. Wiopt, WNC3 Workshop*, Hammanet, Tunisia, May 2014, pp. 569–574.

[7] M. Gregori *et al.*, "Joint transmission and caching policy design for energy minimization in the wireless backhaul link," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 1004–1008.

[8] V. Bioglio *et al.*, "Optimizing MDS codes for caching at the edge," in *Proc. IEEE GLOBECOM*, San Diego, CA, CA, USA, Dec. 2015, pp. 1–6.

[9] K. Poularakis *et al.*, "Multicast-aware caching for small-cell networks," in *Proc. IEEE World Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 2300–2305.

[10] K. Poularakis, *et al.*, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[11] D. Lopez-Perez *et al.*, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 22–30, Jun. 2011.

[12] L. Breslau *et al.*, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1. New York, NY, USA, Mar. 1999, pp. 126–134.