Collaborative Caching for Multicell-Coordinated Systems

Abdallah Khreishah
Electrical and Computer Engineering Department
New Jersey Institute of Technology
Newark, NJ 07102
abdallah@njit.edu

Jacob Chakareski
Electrical and Computer Engineering Department
The University of Alabama
Tuscaloosa, AL 35487
jacob@ua.edu

***Abstract*—Cellular networks have become a major Internet gateway over which online video is increasingly accessed. Data caching in cellular networks brings the accessed content closer to the requesting clients, thus enhancing simultaneously the performance of the video application and the operational efficiency of the cellular network. State-of-the-art caching schemes for cellular networks either employ traditional approaches such as Least Recently Used (LRU) caching or optimize the cache placement at each base station independently. We study the problem of collaborative content caching among base stations such that their aggregate operational cost is minimized or the profit earned by the service provider is maximized, given their caching capacity constraints. We distinguish between two cases: (i) non-coded data in which a content item is either stored at a base station or not, and (ii) coded data, where segments of the fountain or network coded content item are stored at multiple base stations. For the non-coded case, we derive an integer programming formulation and prove its NP-completeness. We also design a fully polynomial-time approximation algorithm for solving the problem of interest. For the coded case, we derive a linear program formulation that can be solved in polynomial time. Our simulation results show that the proposed collaborative caching schemes provide considerable advances over non-collaborative competitors.**

***Index Terms*—Multicell-coordinated systems, caching, collaborative caching, cellular networks, heterogeneous networks.**

I. Introduction

Cellular networks are used extensively to access the Internet presently. A recent report by Cisco projects that mobile data volume will grow 11 times during the period 2013-2018 [1]. This traffic growth at the base station will lead to higher energy consumption and packet delay, if no new traffic engineering methods are discovered. Simultaneously, the dramatic data storage cost decrease that is witnessed presently suggests that future base stations will be equipped with data storage capability. Utilizing it to cache online content accessed by wireless users contributes to the objective of bringing the content closer to users, thereby enhancing the application (latency) and network (energy efficiency and throughput) performance at the same time.

Related work on base station caching is limited. Blasco and Gunduz [2] considered the problem of estimating the content popularity at a base station and minimizing the total delay of content retrieval, where the minimization is mapped to a knapsack problem [3]. The study in [4] considered the problem of reducing the delay of content delivery using caching at wireless helper nodes. The helpers are small-cell base stations that have high storage capability and low coverage. In a follow-up work [5], the authors extended their model so that available helpers are differentiated based on their proximity to the served node. Bastug et al. [6] considered optimizing the parameters of a single base station cache. In [7–9], the authors studied hierarchical caching in cellular backhaul networks. Information-theoretic studies of hierarchical caching are carried out in [10, 11].

Note that the data traffic at different base stations exhibits spatial diversity, as the users's access patterns and rates may vary from base station to base station. The studies above do not address the challenges and exploit the opportunities brought forward by this phenomenon, as they do not allow for base station collaboration. Only recently, Wang et al. [12] considered a heuristic collaborative caching scheme, under the assumption that the content popularity is constant across base stations, thereby limiting the prospective benefits of their approach. The work in [4, 5] also disregards spatial traffic diversity and assumes that a user terminal can communicate to multiple helper nodes, thereby disregarding prospective interference.

In this paper, we propose a collaborative caching scheme that exploits traffic diversity, where the content can be either coded or not. In the non-coded case, each content item is either cached completely at a base station or not at all. We formulate the problem of minimizing the aggregate cost or maximizing the aggregate benefit of delivering the accessed content in this case as an integer program and prove that it is NP-complete. We also develop a near-optimal polynomial-time algorithm based on dynamic programming. For the coded data case, we consider caching fountain or network coded packets [13] of the content items at different base stations. Here, we formulate the problem of interest as a linear program. In our experiments, we demonstrate significant operational efficiency gains relative to state-of-the-art non-collaborative methods. Furthermore, relative to [4, 5], we provide much stronger performance approximation guarantees ($(1-\epsilon)$ compared to 1/2). We consider that a user can communicate to its own base station only, thus, interference, which is disregarded in [4, 5], does not represent an issue in our case.

II. Problem Setup

In multi-cellular systems, the cell base stations are interconnected via back-haul links that also serve as the access network to the Internet, as illustrated in Figure 1. Note that this model is quite general as it includes the Long Term Evolution (LTE) system and the community WiFi network. The above collaborative model represents the actual industrial model for cellular networks, where the base stations are owned by a company and each wireless carrier can lease a part of the base station. However, the wireless carrier is responsible for paying the energy bill for its share of the base station. Though we focus on cellular networks in the paper, this setting can be applied to other wireless access networks.

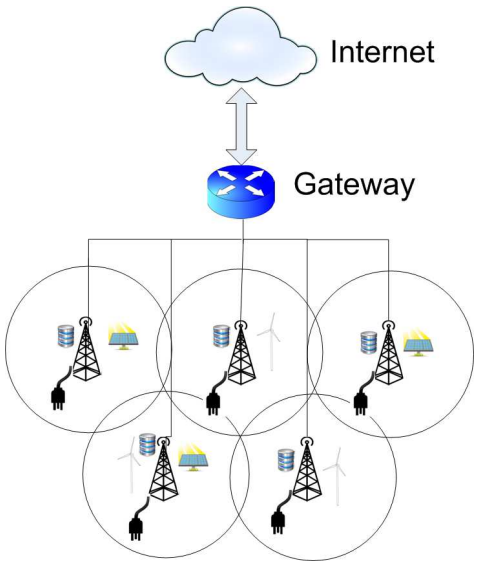

Fig. 1. Cellular access network model.

We consider that there are N content items and K base stations. The base stations can collaborate among them via the backhaul links. The access rate (popularity) of content item j by users at the i-th base station is denoted as a_{ij}. The decision to deliver/serve content j requested by a client at the i-th base station from the cache of the k-th base station is denoted as $X_{ij}^k = 1$, otherwise $X_{ij}^k = 0$. Thus, the action $X_{ij}^i = 1$ represents the event of serving item j locally – from the cache of the i-th base station. Finally, $X_{ij}^{K+1} = 1$ indicates the event of serving content j from the Internet. Throughout the paper the terms client and user are used interchangeably.

We use B_j to represent the size of content j (say in bytes) and C_i to represent the capacity of the cache at the i-th base station. The binary variable W_{ij} is used to specify whether the j-th content is cached at the i-th base station or not. Serving a content item locally, from another base station, or the Internet will incur different levels of transmission cost to the service provider, of progressively increasing magnitude. Simultaneously, the customer satisfaction level (of the provided service) will decrease as the requested content item is delivered from a farther location[1], as the delivery latency will progressively increase with distance.

Note that customer satisfaction is uniquely related to profit earned by the provider, as less satisfied customers will tend to use the service less often and cancel their service subscription sooner (thereby leading to *lower profit* for the provider). On the other hand, more satisfied customers will tend to use the service more often and retain their service subscription for longer periods of time (thereby leading to *higher profit* for the provider). Therefore, minimizing the overall operational cost of the service provider or maximizing its earned profit is equivalent in this context. Our analysis henceforth will focus on the latter optimization instance, but as we will show, our result can be easily modified to work for both the maximization and minimization instances.

Formally, let R_i denote the reward (profit) earned by the provider if a content item requested at base station i is served locally, i.e., directly from its cache. We use R_i^k to denote the profit earned by serving a content item requested at base station i from the cache of base station $k = 1, .., K$, where R_i^{K+1} denotes the earned profit when the content item is served from a remote Internet server. We are interested in filling the base station caches with content items such that the aggregate reward earned by the provider is maximized. Given the considered setup, we assume $R_i \geq R_i^k > R_i^{K+1}$.

III. Formulation

The problem of interest without coding can be formulated as follows:

$$\max \sum_i R_i(\sum_j a_{ij} X_{ij}^i) + \sum_i \sum_{k \neq i} R_i^k (\sum_j a_{ij} X_{ij}^k) + \sum_i R_i^{K+1} (\sum_j a_{ij} X_{ij}^{K+1})$$

subject to:

$$\sum_j W_{ij} B_j \leq C_i, \forall i \quad (1)$$

$$X_{ij}^k \leq W_{kj}, \forall i, j, k \quad (2)$$

$$\sum_{k=1}^{K+1} X_{ij}^k = 1_{\{a_{ij} \geq 0\}} \forall i, j \quad (3)$$

The decision variables are X_{ij}^k and W_{ij}. Thus, we seek to decide the content to cache at each base station and the cache from which we deliver a given content item to a client, such that the objective function is maximized. The latter comprises three terms. The first represents the overall reward obtained when the requested content is found cached at the local base station of a requesting client. The second represents the reward when the content is retrieved from the cache of another base station and the third term represents the reward obtained when the content is retrieved from the Internet.

The first set of constraints represents the base station cache capacity constraints. The second set states that we cannot

[1]The local base station being the closest; The Internet server the remotest.

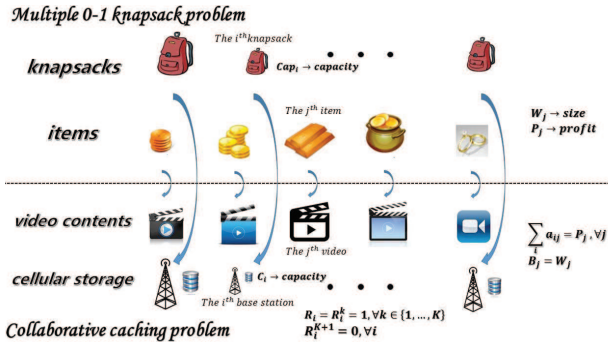

Fig. 2. The reduction performed in the proof of NP-hardness.

retrieve the content from a base station, unless it is cached there. The last set of constraints states that a client should get exactly one copy of the requested content.

Lemma 1: The above problem is NP-complete.

Proof: We can check that a given solution is feasible by verifying that (1)-(3) are met. This can be done in polynomial-time. What remains to show is that our problem is NP-hard.

We perform a mapping from the well-known 0-1 multiple knapsack problem [3]. The problem is defined in [3] as follows: Given a set of N items and K knapsacks, $K \leq N$, with P_j and W_j the profit and the weight of the j-th item, respectively, and Cap_i the capacity of the i-th knapsack, select K disjoint subsets of items such that the total profit due to selected items is maximized, and each subset can be assigned to a different knapsack whose capacity is no less than the total weight of items in the subset. We perform the mapping to an instance of our problem as follows: (i) The i-th knapsack is mapped to the cache of the i-th base station with $C_i = Cap_i$. (ii) The j-th item is mapped to the j-th content with $B_j = W_j$ and $\sum_i a_{ij} = P_j, \forall j$. (iii) We set $R_i = R_i^k = 1, \forall k \in \{1, \ldots, K\}$ and $R_i^{K+1} = 0$.

Note that a solution $\mathcal{A}$ to the 0-1 multiple knapsack problem is feasible with value $\mathcal{V}$ iff it is a feasible solution to the instance of our problem with value $\mathcal{V}$. Note also that the reduction can be performed in polynomial-time. Hence, our problem is NP-complete. ■

Note that our original problem allows item j to be cached at more than one base station. However, due to setting the profits $R_i = R_i^k = 1$ in the instance used in the proof, any solution that involves caching item j at different base stations, can be converted to another one that caches the content only at one base station and achieves the same reward. We can prove that the minimization version of the problem is NP-hard using a similar reduction with $R_i = R_i^k = 0, k \leq K$, and $R_i^{K+1} = 1$. Figure 2 illustrates the problem reduction carried out as part of the proof of NP-hardness.

IV. Polynomial-Time Approximation Scheme

We formulate a polynomial-time approximation algorithm for solving the problem of interest based on dynamic programming [14]. We assume here that R_i^k is the same for all $i \neq j$ combinations. Let us denote $\bar{R} = R_i^k$, $\forall i \neq k$. Therefore, the objective function $O(\{X_{ij}^k\})$ can be modified as follows:

$$\sum_{i,j} R_i a_{ij} X_{ij}^i + \sum_{\substack{i,j,k:k\neq i,\\ k\leq K}} \bar{R} a_{ij} X_{ij}^k + \sum_{i,j} R_i^{K+1} a_{ij} X_{ij}^{K+1}.$$

Thus, we can write our optimization problem as

$$\max_{\{X_{ij}^k\}} O(\{X_{ij}^k\}); \quad \text{s.t ((1)-(3))}. \tag{4}$$

If we scale the profit values associated with caching content item j at base station i such that they are polynomially bounded in KN (the number of decision variables), we can solve the scaled instance via dynamic programming and return an approximative solution to the original problem. Furthermore, if we adjust the scaling such that it integrates the approximation factor ϵ accordingly, then the resulting algorithm will also be polynomial-time with respect to $1/\epsilon$.

Now, let $w_{ij} = R_i a_{ij} + \sum_{k\neq i} \bar{R} a_{kj}$ represent the maximum profit that can be earned by caching item j at base station i. We select $w^{\max} = \max_{i,j} w_{ij}$ to be our scaling factor. Then, we define $p = \lfloor \log(\epsilon w^{\max}/NK) \rfloor$ to be a precision parameter . Let $\mathcal{C} = \{1, \ldots, K\} \times \{1, \ldots, N\}$ define the vector product set of possible data units and caches. Then, $C = |\mathcal{C}|$ and $c = (i, j) \in \mathcal{C}$ define the set size and a member element. Note that the re-indexing will allow us to consider (4) conceptually as a knapsack problem over $\mathcal{C}$, with multiple (knapsack – caching capacity) constraints affiliated with the base stations. In brief, if $w^{\max}$ represents the maximum reward that can be earned by caching a single item $c \in \mathcal{C}$, then $|\mathcal{C}|w^{\max}$ represents the upper bound on aggregate profit. Let $S_{i,p}$ denote a subset of $\{c_1, \ldots, c_i\}$ that takes the smallest caching space and exhibits the highest profit, $\forall i \in \{1, \ldots, |\mathcal{C}|\}$ and $p \in \{1, \ldots, |\mathcal{C}|w^{\max}\}$. Let $A(i, p)$ denote the size of the set $S_{i,p}$. Using the dynamic programming recurrence relation, $A(i + 1, p) = \min\{A(i, p), \text{Size}(c_{i+1}) + A(i, p - \text{Reward}(c_{i+1}))\}$, we can populate $A(i, p)$, $\forall i, p$. The optimal caching configuration then corresponds to the set $S_{i,p}$ that maximizes p and exhibits the smallest $A(i, p)$ that does not violate the caching capacity constraints. Our algorithm computes this set efficiently.

We first present our approximation algorithm and then prove its approximation guarantees and fully-polynomial-time nature. The algorithm is described in Algorithm 1. It utilizes a data structure $\mathbf{Q}_l$ that keeps track of the explored state-space (the content items to be cached and the resulting profit) through stage l. That is, the member elements $(\mathbf{T}, P) \in \mathbf{Q}_l$ correspond to subsets ($\mathbf{T}$) of size $k = 1, ...l$ of the first l elements in $\mathcal{C}$ that feature the maximum earned reward (P) for the given cached data volume ($\sum_{i,j:c\in\mathbf{T}} B_j$). For every subsequent l ($c \in \mathcal{C}$), the algorithm comprises an expansion phase, where the optimal paths (subsets of cached data) maintained in $\mathbf{Q}_{l-1}$ are branched out by considering the next decision variable (to cache item j at base station i), and a pruning phase, where only optimal paths are maintained after the expansion. The caching capacity constraints are maintained during the expansion phase. At the completion of stage KN, the algorithm terminates by selecting the caching configuration

$\mathbf{T}^*$ in $\mathbf{Q}_{KN}$ that exhibits the maximum profit P^*. In the context of the notation used in (4), if $c \in \mathbf{T}^*$, then $X_{ij}^i = 1$ and $X_{kj}^{K+1} = 0, \forall k$, where $(i,j) \leftarrow c$. Next, if $\exists i \leq K : X_{ij}^i = 1$ and $\exists k \neq i : X_{kj}^m = 0, m \leq K$, we set $X_{kj}^i = 1$. Finally, $\forall i,j : X_{ij}^k = 0, k \leq K$, we set $X_{ij}^{K+1} = 1$.

Algorithm 1 Fully Polynomial-Time Approximation

1: Initialize $\mathbf{T} = \emptyset, P_0 = 0, \mathbf{Q}_0 = \{(\mathbf{T}, P_0)\}$
2: **for** $\forall c \in \mathcal{C}, l = 1$ to C **do**
3: $(i,j) \leftarrow c$
4: Grow $\mathbf{Q}_l = \mathbf{Q}_{l-1} \cup \{(\mathbf{T} \cup \{c\}, P_{l-1} + w'_{ij}) | \sum_{c' \in \mathbf{T}: i'=i} B_{j'} + B_j \leq C_i, (\mathbf{T}, P_{l-1}) \in \mathbf{Q}_{l-1}\}$

where $w'_{ij} = \begin{cases} \lfloor \frac{R_i a_{ij}}{10^p} \rfloor + \sum_{k \neq i} \lfloor \frac{\bar{R} a_{kj}}{10^p} \rfloor, & \text{if } \nexists c' \in \mathbf{T} : j' = j, \\ \lfloor \frac{R_i a_{ij}}{10^p} \rfloor - \lfloor \frac{\bar{R} a_{ij}}{10^p} \rfloor, & \text{otherwise.} \end{cases}$

5: **if** $\exists (\mathbf{T}^1, P^1), (\mathbf{T}^2, P^2) \in \mathbf{Q}_l : P^1 = P^2$ and $\sum_{c \in \mathbf{T}^1} B_j > \sum_{c \in \mathbf{T}^2} B_j$ **then**
6: Prune $\mathbf{Q}_l = \mathbf{Q}_l \setminus \{(\mathbf{T}^1, P^1)\}$
7: **end if**
8: **end for**
9: Select $\{(\mathbf{T}^*, P^*)\} \in \mathbf{Q}_C : P^* = \max_P \{(\mathbf{T}, P) \in \mathbf{Q}_C\}$

Theorem 1: Let OPT denote the maximum value of the objective in (4) and let $\{X_{ij}^k\}^*$ denote the corresponding solution. The solution $\{X_{ij}^k\}'$ computed by Algorithm 1 satisfies

$$O(\{X_{ij}^k\}') \geq (1-\epsilon) \cdot \text{OPT}.$$

Proof: In Algorithm 1, we effectively scale down (and then round) the reward earned by each item-pair $c = (i,j)$ by a factor $S = \epsilon w^{\max}/NK$. Thus, the reward w'_{ij} earned by including an item c in the scaled instance will satisfy $S \cdot w'_{ij} \leq w_{ij}$. This in turn implies that the reward earned by $\{X_{ij}^k\}^*$ can drop at most S for every cached item (according to $\{X_{ij}^k\}^*$), when evaluated in the scaled instance. Therefore, we can bound the overall reward drop as

$$O(\{X_{ij}^k\}^*) - S \cdot O'(\{X_{ij}^k\}^*) \leq CS, \tag{5}$$

where O' denotes the objective in (4) for the scaled instance.

Now, Algorithm 1 computes $\{X_{ij}^k\}'$ using dynamic programming. Thus, $\{X_{ij}^k\}'$ represents the optimal solution to the scaled instance of (4). Hence, $O'(\{X_{ij}^k\}') \geq O'(\{X_{ij}^k\}^*)$ must hold. Using this, we can write the following set of inequalities

$$\begin{aligned} O(\{X_{ij}^k\}') \geq S \cdot O'(\{X_{ij}^k\}') &\geq S \cdot O'(\{X_{ij}^k\}^*) && (6) \\ &\geq O(\{X_{ij}^k\}^*) - CS && (7) \\ &= \text{OPT} - \epsilon w^{\max} && (8) \\ &\geq (1-\epsilon) \cdot \text{OPT} && (9) \end{aligned}$$

where (8) follows from (5) and (9) holds as $\text{OPT} \geq w^{\max}$. ■

The running time of Algorithm 1 is polynomial in C, because it corresponds to completing a table $A(c,r)$ that comprises $C^2 \lfloor w^{\max}/S \rfloor$ entries, i.e., $c = 1, \ldots, C$ and $r = 1, \ldots, C\lfloor w^{\max}/S \rfloor$. Note that the scaling allows Algorithm 1 to be fully polynomial, i.e., its running time is polynomial in $1/\epsilon$ as well, given that $w^{\max}/S = C/\epsilon$. A minimization version of Algorithm 1 requires reversing the inequality in Line 5. The variables w_{ij} will then represent the cost of caching item j at base station i.

V. The Coded Case

Our problem is NP-hard as the decision variables are binary (to cache or not). Coding the content packets reduces the problem's complexity. We consider intrasession network coding where only packets of the same content are coded together. The same discussion applies if fountain codes are used. If a content item comprises packets $p_1, \ldots, p_n$, we generate coded packets $D_u = \sum_{v=1}^n \beta_v p_v$, where $\beta_v, \forall v$ are random coefficients and the arithmetic operations are performed over a finite field.

Lemma 2: If the reward of retrieving coded content from multiple caches is the weighted sum of the reward attained from each of the caches, such that the weight given to each cache is proportional to the number of coded packets retrieved from that cache, the new reward maximization problem can be formulated in the same way as our original problem after relaxing the integer variables X_{ij}^k and W_{ij} to be fractional variables. In this case, the optimal solution can be obtained using a linear program in polynomial-time.

Proof: We redefine X_{ij}^k to be the fraction of the j-th content packets served from the cache of the k-th base station to the users at the i-th base station and W_{ij} to be the fraction of the j-th content packets stored at the i-th base station. Thereby, the objective function of our problem satisfies the definition of the total reward with network coding in the statement of this lemma. What remains is to show that the constraints still hold after relaxing the integer variables.

Note that here $W_{ij}B_j$ represents the size of the coded packets for the j-th content cached at the i-th base station. Thus, (1)-(2) apply straightforwardly. (3) prevents duplicating item j at base station i. This is ensured by design with network coding, as any n different coded packets will be linearly independent with a very high probability, if the coding coefficients β_v are chosen uniformly at random from a finite field [15]. Thus, (3) states here that any n packets retrieved from any base station(s) should suffice to reconstruct item j. ■

The lemma can be extended to the minimization case.

VI. Numerical Results

We compare four caching schemes: LRU, optimal non-collaborative caching [2], and our collaborative technique with and without coding. Since obtaining real cost data was easier, we study performance relative to cost minimization here, relating caching cost to the associated energy consumption of serving the content. Concretely, if item j is cached locally at base station i, then the energy consumption cost will reflect the base station transmission only. If j is retrieved from another base station k, the consumed energy will represent the sum of the energy consumed to move the content between i and k, and the energy consumed by i to transmit j. Note that the energy required to move content between base stations is proportional to the number of hops the content has to travel. We account for the cost of retrieving j from an Internet server similarly. Based on [16], we assume that j will traverse 15 hops, if

retrieved from the Internet. The values of energy consumption by wireless and wireline transmissions are taken from [17, 18], such that the energy per bit consumed by wireless transmission is 3.5 μJoule and the energy per bit consumed by wireline transmission is 0.5 μJoule per hop. N and K are set to 10^4 and 15, respectively. We consider that the base stations interconnect via a back-haul link, which is further linked to the Internet. B_j (in MB) is selected uniformly at random from the discrete set $\{10, 11, \ldots, 20\}$ and the number of users at each base station is selected in the same way, i.e., from a uniform distribution. The content access rates at each base station are modeled as independent Zipf distributions, as observed in prior studies [19], with parameter $\gamma = 0.8$. The cache size at each base station is set to 5GB.

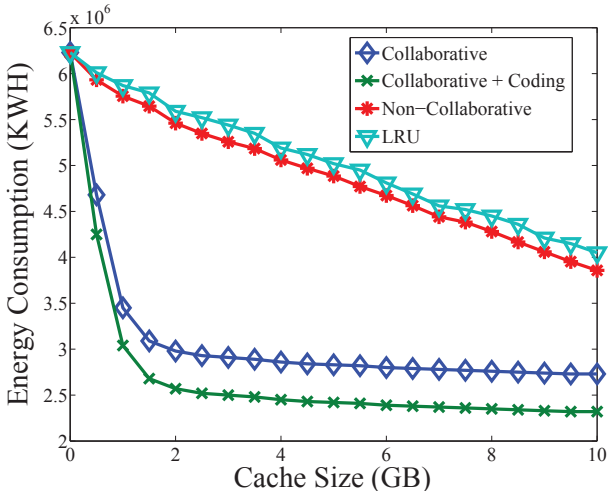

Fig. 3. Energy cost versus cache size.

Fig. 3 shows the energy consumption of the system under each of the caching schemes, for the given cache size at a base station. We can see that the cost savings due to collaboration increase with the cache size up to a specific value and after that they saturate. Collaboration based on coded data provides the best performance. At cache size of 2 GB, the coded instance of our approach cuts energy consumption by more than 45% compared to the other schemes under comparison.

Fig. 4 examines the energy consumption of the system as a function of the number of base stations. We can see that our techniques provide significant cost savings ranging from 50%-90% as the number of base stations increases.

In Fig. 5, we show results that examine the relation between energy consumption and γ. We can see that both of our techniques consistently outperform their competitors with a wide margin, over a broad range of γ values (diverse content popularity distributions).

Finally, in Fig. 6 we examine the empirical CDF of the normalized (w.r.t. LRU) energy consumption of the system under the three other caching schemes. Again, it can be seen that the collaborative techniques provide significant energy savings relative to the non-collaborative method and LRU.

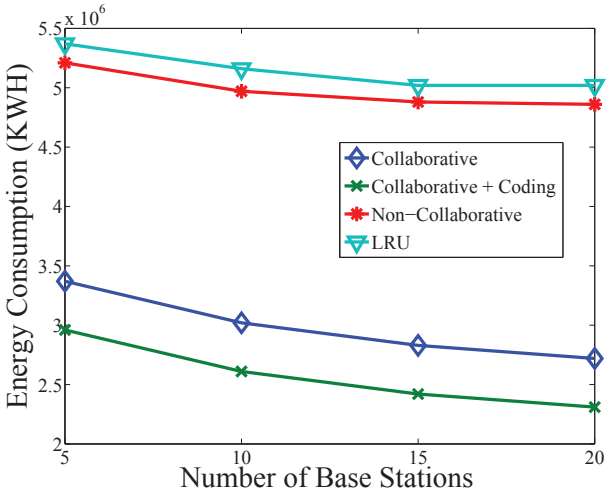

Fig. 4. Energy cost versus # base stat.

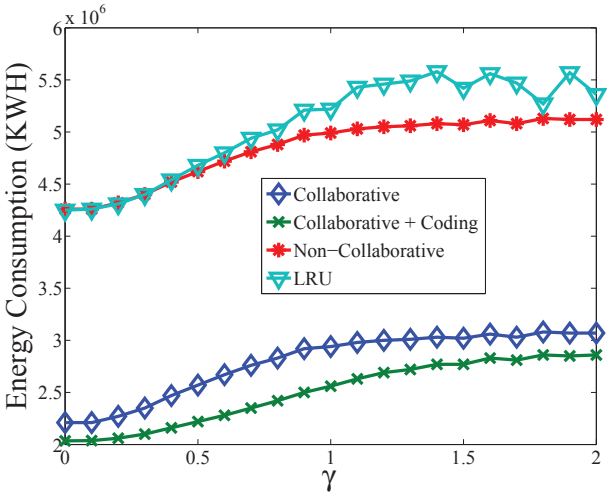

Fig. 5. Energy cost versus γ.

VII. Conclusion

We studied the problem of optimizing the performance of a collaborative multi-cell system equipped with data storage. Unlike related work, our approach exploits the spatial traffic diversity at different base stations and the opportunities for inter-base station collaboration. We considered coded-data and non-coded data instances of this problem in our analysis. The non-coded instance is formulated as an integer program, which is proven to be NP-complete. We also designed a near-optimal fully polynomial algorithm for this case. The problem in the coded-case is formulated as a linear program that can be solved in polynomial-time. Our experiments show the benefits of collaborative caching over optimal non-collaborative and state-of-the-art heuristic methods.

References

[1] "Cisco visual networking index." [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

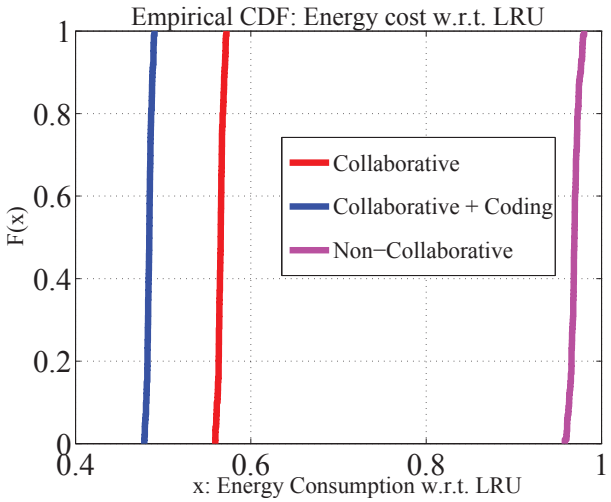

Fig. 6. Energy cost CDF (w.r.t. LRU).

[2] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," *arXiv preprint arXiv:1402.3247*, 2014.

[3] S. Martello and P. Toth, *Knapsack problems*. Wiley New York, 1990.

[4] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *IEEE INFOCOM*, 2012.

[5] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," *IEEE Trans. Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[6] E. Bastug, M. Bennis, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," *arXiv preprint arXiv:1405.3477*, 2014.

[7] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, "To cache or not to cache: The 3G case," *Internet Computing, IEEE*, vol. 15, no. 2, pp. 27–34, 2011.

[8] H. Ahlehagh and S. Dey, "Video caching in radio access network: impact on delay and capacity," in *IEEE WCNC*, 2012.

[9] P. Ostovari, A. Khreishah, and J. Wu, "Cache content placement using triangular network coding," in *IEEE WCNC*, 2013.

[10] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *Proc. IEEE Int'l Symp. Inform. Theory*, July 2014.

[11] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," in *Proc. IEEE Int'l Symp. Inform. Theory*, July 2013.

[12] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[13] A. Shokrollahi, "Raptor codes," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2551–2567, 2006.

[14] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[15] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, 2006.

[16] A. Fei, G. Pei, R. Liu, and L. Zhang, "Measurements on delay and hop-count of the internet," in *IEEE GLOBECOM98-Internet Mini-Conference*, 1998.

[17] J. Huang, F. Qian, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *ACM MobiSys*, 2012.

[18] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell, "Profiling per-packet and per-byte energy consumption in the NetFPGA gigabit router," in *IEEE (INFOCOM WKSHPS)*, 2011.

[19] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE INFOCOM*, 1999.