Weighted Network Traffic Offloading in Cache-enabled Heterogeneous Networks

Xiuhua Li, Xiaofei Wang, and Victor C. M. Leung
Dept. Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
Email: {lixiuhua, xfwang, vleung}@ece.ubc.ca

***Abstract*—Due to explosive demands of multimedia services from mobile users, the growing network traffic load becomes a severe challenge for mobile network operators (MNOs). To address this problem, content caching is regarded as an effective emerging technique to reduce the duplicated transmissions of the content downloads demanded by mobile users, while heterogeneous networks (HetNets) are regarded as an effective technique to increase the network throughput. Thus, this paper focuses on content caching in HetNets to offload the weighted network traffic, in which we consider the problem of minimizing the weighted expected sum of traffic load of accessing the requested contents. By transforming the irregular problem into a binary integer linear programming problem, we propose a novel suboptimal heuristic algorithm with polynomial-time complexity to solve the problem, instead of using the existing optimal branch-and-bound method with exponential-time complexity. Numerical results demonstrate that our proposed content caching framework can reduce the weighted expected sum of traffic load significantly.**

I. Introduction

With the fast advances of wireless communication technologies, mobile users are increasingly enjoying multimedia services on mobile terminals (e.g., smart phones and tablets) by downloading multimedia contents (e.g. audio and video) from the Internet via mobile networks, which is contributing to the exponential growth of traffic in the current mobile networks. How to handle this traffic, especially on the radio access network and the backhaul networks, is a critical concern of mobile network operators (MNOs) [1]–[4], which will need to be addressed via advances in network architectures and data transmission technologies of the next generation, i.e., 5G mobile networks.

On one hand, one emerging technique is to cache popular contents at mobile network edges to reduce the traffic due to a massive number of duplicated content downloads via the backbone networks and cellular links [5], [6]. Since content popularity follows the "Power Law" (e.g., the top 10% of videos in YouTube account for about 80% of all the views [7]), contents can be cached inside mobile fronthaul and backhaul networks and then mobile users' requests for the same contents can be satisfied without multiple transmissions from service providers (SPs) outside the MNO's networks. As a result, duplicated traffic load can be greatly decreased in cache-enabled mobile networks.

On the other hand, heterogeneous networks (HetNets) are regarded as an effective technique to increase the network throughput. In a HetNet, the nodes are densely deployed, including macro base stations (BSs), micro BSs, pico BSs (PBSs), femto BSs (FBSs) and relays. Thus, the distance between BSs/relays and users is reduced, which can improve the area spectral efficiency as well as network capacity [8]. However, the exponential growth of network traffic also requires the high-speed backhaul for the connection of different type of BSs/relays and SPs [9].

Thus, combining content caching and wireless HetNets together, i.e., cache-enabled HetNets, can effectively reduce the network traffic. Since the contents owned by SPs are increasing explosively and caching all the contents is impractical, proper caching strategies are crucial. Specifically, many factors should be considered to decide whether a content should be cached, e.g., its popularity, storage size, locations of existing replicas over the network topology and so on [3]. There have been several studies focusing on caching popular contents at network devices in proximity to users to reduce traffic load over the backbone networks of MNOs [5], [10]. Besides, the proposed FemtoCache in [11] and AMVS-NDN in [12], focus on caching popular contents in BSs, evolved NodeBs (eNBs) or femtocells to reduce the traffic load and increase the number of served users. To reduce the traffic load and improve users' quality of service (QoS), especially on the delay of accessing users' requested contents, cooperative cell caching schemes at the BSs were studied in [1], [3], while video caching approaches in radio access networks (RANs) were proposed in [13]. The concept of "Caching-as-a-Service" (CaaS), a caching virtualization framework was proposed in [4] along with the development of Cloud-based RANs (C-RANs) and the virtualization of Evolved Packet Core (EPC), aiming to reduce inter-MNO and intra-MNO traffic loads. In [9], based on stochastic analysis, cache-based content delivery is proposed and analyzed in a three-tier HetNet, where BSs, relays and device-to-device (D2D) pairs are included, aiming to offload the traffic and improving users' QoS. However, these studies have not considered the fact that the traffic loads between different network devices have different weights or operational costs for MNOs.

Therefore, in this paper, we are motivated to focus on content caching in HetNets to offload the weighted network traffic. This problem is formulated as a minimization of the weighted expected sum of traffic loads of accessing the requested contents under the constraints of the cache sizes and backhaul link rates in a HetNet. By exploring the properties of the optimization problem, the irregular problem is transformed

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

into a binary integer linear programming (BILP) problem, which we propose to solve using a novel suboptimal heuristic algorithm that has polynomial-time complexity, rather than the conventional branch-and-bound (BNB) method that suffers from exponential-time complexity.

The remainder of this paper is organized as follows. We describe the system model and formulate the problem in Section II. In Section III, we propose a low-complexity iterative greedy heuristic caching method and make some bound analysis. Numerical results are shown in Section IV to evaluate the proposed caching strategies. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

A. System Model

A cache-enabled multi-cell HetNet system model is shown in Fig. 1[1]. Outside the MNO network, there are some SPs (e.g., YouTube, Facebook, and so on) offering multimedia contents over the Internet while inside the MNO network, there are a great number of cells covering the whole service area. In each cell, its macro base station (MBS) is connected with several geographically dispersed pico base stations (PBSs) through cables or optical fibers with limited speed or capacity. We assume that a user in each cell is associated with the local PBS closest in distance as well as the local MBS, while a PBS can serve multiple local users.

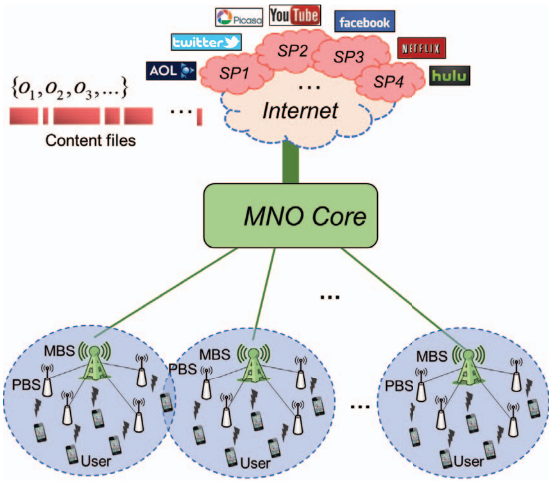

Fig. 1. Cache-enabled system model in HetNets.

Specifically, MBSs have no caching functionality while PBSs are able to cache some contents with limited cache storage capacity to satisfy users' content requests and reduce duplicated content transmissions and thus network backhaul traffic load. However, MBSs can decide how to effectively select popular contents to cache in their connected PBSs, and can properly update the PBSs' contents. Besides, a MBS maintains a list of all the connected PBSs' cached contents at the cost of a small traffic overhead that is assumed negligible. Moreover, we assume that the content popularity changes slowly. For instance, popular but short news videos with short lifetimes are updated every few hours while new movies and new music videos with long-lifetime are posted weekly or monthly. To reduce the traffic load and avoid possible network traffic congestion especially in busy hours, popular contents especially those with long lifetimes can be cached in peak-off hours (e.g., late night). Since the content popularity can be regarded as fixed over a relatively long time [1], [11], the cost of updating the PBSs' contents can be neglected. We assume that the content popularity can be obtained in advance or predicted by the system through learning or analysis of users' behavior and preference, and thus the content popularity in different PBSs is known for the associated MBS. Once the optimal content caching strategy is achieved, MBSs centrally push the contents to the connected PBSs via backhaul links.

In the system model, we mainly consider the downlink transmissions of the system. Once a user's content request is generated, the user's local PBS satisfies the request if the content is cached there. Otherwise, if the content is cached in another PBS, then the MBS helps to deliver the content from the PBS caching the content to the local PBS via the backhaul link. Finally, if the content is not cached in the cell, then the MBS directly handles the user request by downloading the content from the Internet via a backhaul link of the MNO's core network and delivering it to the user via a wireless link. Thus, assuming that the traffic overhead used for establishing the link to deliver the content to a user is very small and can be neglected, to satisfy a user's content request, the traffic load can be generated and divided into four possible cases as: 1) **the traffic load from the local PBS to a user** when the user's requested content is cached in either the local PBS or another PBS in the cell; 2) **the traffic load between two PBSs** when a user's requested content is not cached in the local PBS but cached in another PBS in the cell; 3) **the traffic load from the MBS to a user** when a user's requested content is not cached in the cell and the content request is directly handled by the MBS; 4) **the traffic load outside the MNO network** generated by the MBS to download a content from the Internet via the MNO's core network when the content is not cached in the corresponding cell.

In this paper, we only consider the case of a single cell, equipped with a MBS and M PBSs to serve a great number of users. The storage capacities of the PBSs' caches are denoted by $\{S_1, S_2, \ldots, S_M\}$. There is a catalog of N popular contents, denoted by $\{o_1, o_2, \ldots, o_N\}$. From a practical perspective, contents are assumed to have variable sizes denoted by $\{s_1, s_2, \ldots, s_N\}$. Besides, each content requires a minimum transmission rate of $\{c_1, c_2, \ldots, c_N\}$, while the maximum backhaul link rate between a PBS and the MBS is limited by $\{C_1, C_2, \ldots, C_M\}$. The caching strategy in the single-cell system is denoted by a binary matrix $\mathbf{X} := (x_{ij})_{M\times N} \in \{0,1\}^{M\times N}$, where '1' means caching while '0' means no caching. In other words, a content is assumed to be either entirely cached or not cached. Moreover, the above four cases of traffic load have different constant weighted factors $\{\omega_1, \omega_2, \omega_3, \omega_4\}$, denoting the corresponding expected cost per unit traffic load, in order to satisfy a user's content request. The detailed modeling parameters and notations are given in Table I. Here, we assume $\omega_4 > \omega_3 > (\omega_1 + \omega_2)$.

TABLE I
MODELING PARAMETERS AND NOTATIONS

Symbol	Definition
M	Total number of PBSs in the single-cell system
N	Total number of popular contents in the network
o_j	Content j
S_i	Size of cache storage of PBS i
s_j	Size of content o_j
C_i	Maximum backhaul link rate between PBS i and the MBS
c_j	Required minimum rate of delivering content o_j
x_{ij}	A decision 0-1 variable for PBS i to cache content o_j
r_{ij}	Popularity for content o_j at PBS i
R_j	Popularity for content o_j overall the system
ω_1	weighted factor w.r.t. the traffic load from the local PBS to a user
ω_2	weighted factor w.r.t. the traffic load between two PBSs
ω_3	weighted factor w.r.t. the traffic load from a MBS to a user
ω_4	weighted factor w.r.t. the traffic load outside the MNO network

Moreover, following [7], we assume that the overall popularity of the content o_j, denoted as $R_j = \sum_{i=1}^{M} r_{ij}$ for $\forall j$, satisfies the Zipf distribution. Specifically, we have

$$R_j = \frac{rank_j^{-\beta}}{\sum_{k=1}^{N} rank_k^{-\beta}}, \quad \forall j \tag{1}$$

where $rank_j$ is the rank of the content o_j in the descending order of content popularity, and the exponent $\beta \in (0, 1]$ is a constant that reflects the skew of the content popularity distribution. Clearly, we have $\sum_{i=1}^{M} \sum_{j=1}^{N} r_{ij} = \sum_{j=1}^{N} R_j = 1$.

B. Problem Formulation

From the perspective of the MNOs, it is essential to reduce the network traffic load as well as the cost. In this paper, our objective is to minimize the weighted expected sum of traffic loads with proper caching strategies, thereby reducing system costs as well.

Thus, with caching strategies, the expected sum of traffic loads from the local PBS to a user w.r.t. the content that is cached in either the local PBS or another PBS in the cell can be calculated as

$$TF_1 = MN \sum_{i=1}^{M} \sum_{j=1}^{N} \big(\bigcup_{k=1}^{M} x_{kj} \big) \cdot r_{ij} \cdot s_j \tag{2}$$

where the indicator $\bigcup_{k=1}^{M} x_{kj} := x_{1j} \oplus x_{2j} \oplus ... \oplus x_{Mj}$ is a 0-1 variable that indicates whether there is a copy of the content o_j cached in any PBS in the cell. $\bigcup_{k=1}^{M} x_{kj} = 1$ if and only if there is any $x_{kj} = 1$ w.r.t. k.

The expected sum of traffic load between two PBSs w.r.t. the content that is not cached in the local PBS but cached in another PBS in the cell can be calculated as

$$TF_2 = MN \sum_{i=1}^{M} \sum_{j=1}^{N} \big[\big(\bigcup_{k=1}^{M} x_{kj} \big) - x_{ij} \big] \cdot r_{ij} \cdot s_j . \tag{3}$$

The expected sum of traffic load from a MBS to a user w.r.t. the content that is not cached in the cell can be calculated as

$$TF_3 = MN \sum_{i=1}^{M} \sum_{j=1}^{N} \big(1 - \bigcup_{k=1}^{M} x_{kj} \big) \cdot r_{ij} \cdot s_j . \tag{4}$$

The expected sum of traffic load outside the MNO network w.r.t. the content that needs to be downloaded by the MBS from the Internet via MNO core can also be calculated as

$$TF_4 = MN \sum_{i=1}^{M} \sum_{j=1}^{N} \big(1 - \bigcup_{k=1}^{M} x_{kj} \big) \cdot r_{ij} \cdot s_j . \tag{5}$$

Thus, considering the limitation of the PBSs' cache sizes and backhaul link rates between PBSs and the MBS, the overall problem of minimizing the weighted expected sum of traffic load in the system can be formulated as

$$\min_{\mathbf{X}} \quad \sum_{l=1}^{4} \omega_l \cdot TF_l \tag{6a}$$

$$s.t. \quad \sum_{j=1}^{N} x_{ij} s_j \le S_i, \quad \forall i, \tag{6b}$$

$$N \sum_{j=1}^{N} \big[r_{ij} \cdot \big(\bigcup_{k=1}^{M} x_{kj} - x_{ij} \big) \cdot c_j \big] \le C_i, \quad \forall i, \tag{6c}$$

$$\bigcup_{k=1}^{M} x_{kj} = x_{1j} \oplus x_{2j} \oplus ... \oplus x_{Mj}, \quad \forall j, \tag{6d}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, \forall j. \tag{6e}$$

Specifically, the constraint in (6c) denotes that the expected sum rate of delivering a content between one PBS and any of the other PBSs with the content's required minimum transmission rate is no greater than the maximum backhaul link rate between the PBS and the MBS. The optimization objective of the problem in (6) can be rewritten as $-\sum_{i=1}^{M} \sum_{j=1}^{N} \big[f_{ij} \big(\bigcup_{k=1}^{M} x_{kj} \big) + g_{ij} \, x_{ij} \big] + TF_0$, where $f_{ij} := (\omega_3 + \omega_4 - \omega_1 - \omega_2) MN r_{ij} s_j > 0$, $g_{ij} := \omega_2 MN r_{ij} s_j > 0$ and $TF_0 := (\omega_3 + \omega_4) MN \sum_{i=1}^{M} \sum_{j=1}^{N} r_{ij} s_j$ are positive constants. Besides, we have $\frac{f_{ij}}{g_{ij}} = \frac{\omega_3 + \omega_4 - \omega_1 - \omega_2}{\omega_2} > 1$ for $\forall i, \forall j$.

Remark 1: The non-caching strategy can be regarded as a special case of caching strategies by setting the cache sizes of the PBSs (i.e., $\{S_i\}_{i=1}^{M}$) to zeros. As a result, we have $x_{ij} = 0$ for $\forall i, j$, and the corresponding weighted expected sum of traffic loads becomes TF_0.

C. Transformation of the Optimization Problem

To solve the problem in (6), the first challenge is to transform the $\bigcup$ operation into a regular form. As in our previous work [3], the $\bigcup$ operation can be equivalently transformed through Theorem 1 as follows.

Theorem 1: For a vector $\mathbf{a} \in \{0,1\}^{n\times 1}$, denote $y = \bigcup_{k=1}^{n} a_k$. Then the $\bigcup$ operation on $\mathbf{a}$ is equivalent to $y = \max_{k\in\{1,2,\ldots,n\}} \{a_k\}$, and is also equivalent to two linear inequality conditions as

$$a_k \le y \text{ for } \forall k \in \{1,2,\ldots,n\}, \text{ and } y \le \sum_{k=1}^{n} a_k. \tag{7}$$

Denote $y_j = \bigcup_{k=1}^{M} x_{kj}$ for $\forall j$ and $\mathbf{y} := (y_j)_{1\times N}$. Based on Theorem 1, the problem in (6) can be equivalently transformed as

$$\min_{\mathbf{X},\,\mathbf{y}} \; z = -\sum_{i=1}^{M}\sum_{j=1}^{N} (f_{ij} y_j + g_{ij}\, x_{ij}) \tag{8a}$$

$$\textit{s.t.} \; \sum_{j=1}^{N} x_{ij} s_j \le S_i, \quad \forall i, \tag{8b}$$

$$\sum_{j=1}^{N} (a_{ij} y_j - a_{ij} x_{ij}) \le C_i, \quad \forall i, \tag{8c}$$

$$x_{ij} \le y_j, \quad \forall i, \forall j, \tag{8d}$$

$$y_j \le \sum_{i=1}^{M} x_{ij}, \quad \forall j, \tag{8e}$$

$$x_{ij} \in \{0,1\}, y_j \in \{0,1\}, \quad \forall i, \forall j \tag{8f}$$

where $a_{ij} := N \cdot r_{ij} \cdot c_j$ for $\forall i, \forall j$, are positive constants. The problem in (8) is a BILP problem and thus NP-complete. Note that the overall constraints in (8d) and (8e) are equivalent to $y_j = \max_{i\in\{1,2,\ldots,M\}} \{x_{ij}\}, \; \forall j$.

III. Solutions and Bound Analysis

A. Iterative Greedy Heuristic Method

To solve a BILP and get its optimal solution, the well-known and widely studied BNB methods[14], [15] can be used. However, BNB methods always have exponential time complexity, and thus it is not practical to use BNB methods to solve large-scale BILPs such as the problem in (8).

From the perspective of practical engineering implementation, we propose a low-complexity iterative greedy heuristic method to solve the BILP in (8). The main procedures of the proposed method are shown as follow.

- **Step 1**: After initialization, choose contents to cache in the cell to determine j^* such that $y_{j^*} = 1$, and then choose which PBSs to cache each chosen content with one copy to determine i^* such that $x_{i^*j^*} = 1$. Specifically, based on the main idea of heuristic algorithms for solving regular 0-1 multi-knapsack problems (MKPs)[15], the pair (i^*, j^*) is determined only when the values of $\sum_{i=1}^{M} f_{ij^*})/(s_{j^*} \cdot \sum_{i=1}^{M} a_{ij^*})$ and $a_{i^*j^*}$ are largest under the constraints in (8b) and (8c), in order to get a smaller value of the optimization objective.
- **Step 2**: Choose PBSs to cache the cached contents with multiple copies to determine k^* for each cached content o_{j^*} such that $x_{k^*j^*} = 1, k^* \ne i^*$. Specifically, in a similar way, the pair (k^*, j^*) is determined only when $g_{k^*j^*}/(s_{j^*} \cdot a_{k^*j^*})$ is largest under the constraints in (8b). Note that since more x_{ij} can achieve the value of 1, the constraints in (8c) still holds and becomes more relaxed, which provides more possibilities to cache some of the remaining uncached contents.
- **Step 3**: Repeat **Step 1** and **Step 2** until no more contents can be cached.

The details of the proposed method are shown in Algorithm 1. Besides, the whole procedure of Algorithm 1 takes polynomial time, i.e., $O(MN\log(MN))$, which is mainly bounded by the sorting.

B. Bound Analysis

By relaxing the binary constrains in (8f) into linear constrains, the problem in (8) can be relaxed as

$$\min_{\mathbf{X},\,\mathbf{y}} \; z_R = -\sum_{i=1}^{M}\sum_{j=1}^{N} (f_{ij} y_j + g_{ij}\, x_{ij}) \tag{9a}$$

$$\textit{s.t.} \;\; \text{the same with (8b), (8c), (8d) and (8e),} \tag{9b}$$

$$0 \le x_{ij} \le 1, 0 \le y_j \le 1, \quad \forall i, \forall j \tag{9c}$$

which is a linear programming (LP) problem. The relaxed problem in (9) can be considered that any content can be divided into several parts and partly cached in several PBSs, which brings a high complexity and challenges in managing the contents at the the MBS and is much less practical than the problem of caching entire contents in PBSs studied in (8).

The LP problem in (9) can be solved with the existing revised simplex method or interior point method with low complexity. Denote the relaxed problem's optimal solution and optimal objective as $(\mathbf{X}_R^*, \mathbf{Y}_R^*)$ and z_R^*, respectively. Thus, z_R^* is the lower bound (LB) of the optimal objective z^* of the problem in (8), i.e., $z^* \ge z_R^*$. Besides, $z^* = z_R^*$ holds if and only if $(\mathbf{X}_R^*, \mathbf{Y}_R^*)$ is a binary solution. We use the derived lower bound to evaluate the effectiveness of the proposed polynomial-time iterative greedy heuristic method.

IV. Evaluation Results

In this section, we evaluate the performance of the weighted expected sum of traffic loads of the caching strategies for small-scale and large-scale contents. In our simulations, the PBSs' cache sizes and the maximum backhaul rates between PBSs and the MBS are constant, i.e., $S_i \equiv S$ and $C_i \equiv C$ for $\forall i$. Each content's overall popularity (R_j) follows Zipf distribution with $\beta = 0.95$ and the popularity of each content in each PBS (r_{ij}) is random in $(0,1]$. Each content's size (s_j) and required minimum transmission rate (c_j) is random in [0.001, 1] Gbit and [0.2, 2] Mbps, respectively. Besides, set $[\omega_1, \omega_2, \omega_3, \omega_4] = [1, 2, 4, 5]$ unit cost/Gbit. Note that for a

Algorithm 1 Iterative Greedy Heuristic Algorithm.

```
Input: $M$, $N$, $\mathbf{F} = (f_{ij})_{M\times N}$, $\mathbf{s} = (s_j)_{j=1}^N$, $\mathbf{S} = (S_i)_{i=1}^M$, $(C_i)_{i=1}^M$, $\mathbf{G} = (g_{ij})_{M\times N}$, $\mathbf{A} = (a_{ij})_{M\times N}$.
Initialize $\mathbf{X} = (x_{ij})_{M\times N} = \mathbf{0}_{M\times N}$, $\mathbf{y} = (y_j)_{j=1}^N = \mathbf{0}_N$, $\lambda_j = (\sum_{i=1}^M f_{ij})/(s_j \cdot \sum_{i=1}^M a_{ij})$ for $\forall j$, $i^* = 1$, $j^* = 1$, $\overline{S}_i = S_i$ and $\overline{C}_i = C_i$ for $\forall i$, a set $\mathcal{J} = \emptyset$.
Sort $\mathbf{A}$ into $\mathbf{B}$ in a descending order for each column and label the original order as $\mathbf{D}$, i.e., $b_{ij} = a_{d_{ij},j}, \forall i, \forall j$.
while $\mathcal{J} \neq \emptyset$ do
    —Procedure 1. [Choose a content to cache with one copy]
    Sort $(\lambda_j)_{j=1}^N$ into $(\beta_j)_{j=1}^N$ in a descending order and label the original order as $(\pi_j)_{j=1}^N$, i.e., $\beta_j = \lambda_{\pi_j}, \forall j$.
    for $j = 1$ to $N$ do
        Set $j^* = \pi_j$;
        for $i = 1$ to $M$ do
            Set $i^* = d_{ij}$;
            if $s_{j^*} \leq \overline{S}_{i^*}$ and $a_{kj^*} \leq \overline{C}_k$ for $\forall k \neq i^*$ then
                Set $y_{j^*} = 1$, $x_{i^*j^*} = 1$, $\mathcal{J} \leftarrow \mathcal{J} \cup \{(i^*, j^*)\}$, $\overline{S}_{i^*} \longleftarrow \overline{S}_{i^*} - s_{j^*}$.
                for $k = 1$ to $M$ do
                    Set $\overline{C}_k \longleftarrow \overline{C}_k - a_{tj^*} \cdot (y_{j^*} - x_{kj^*})$.
                end for
                break;      / / Break the loop “for $i = 1$ to $M$”
            end if
        end for
    end for
    —Procedure 2. [Check $\mathcal{J} = \emptyset$]
    if $\mathcal{J} == \emptyset$ then
        break;      / / Break the loop “while”
    end if
    —Procedure 3. [Cache a cached content with multiple copies]
    Set $J = |\mathcal{J}|$, $(\theta_t)_{t=1}^{MJ} = \mathbf{0}_{MJ\times 1}$, $\bar{\mathbf{s}} = (\bar{s}_j)_{j=1}^J$, $\overline{\mathbf{G}} = (\overline{g}_{ij})_{M\times J}$, $\overline{\mathbf{A}} = (\overline{a}_{ij})_{M\times J}$, where for each $j$-th element $(k_i, k_j)$ in $\mathcal{J}$, $\bar{s}_j = s_{k_j}$, $\overline{\mathbf{G}}(:, j) = G(:, k_j)$ and $\overline{\mathbf{A}}(:, j) = A(:, k_j)$.
    for $j = 1$ to $J$ do
        Set $\overline{g}_{k_i,j} = 0$.
        for $i = 1$ to $M$ do
            Set $t = (j-1) \cdot M + i$, $\theta_t = \overline{g}_{ij}/(\bar{s}_j \cdot \overline{a}_{ij})$.
        end for
    end for
    Sort $(\theta_t)_{t=1}^{MJ}$ into $(\phi_t)_{t=1}^{MJ}$ in a descending order and label the original order as $(\varepsilon_t)_{t=1}^{MJ}$, i.e., $\phi_t = \theta_{\varepsilon_t}, \forall t$.
    for $t = 1$ to $(M-1)J$ do
        Set $i^* = \text{mod}(\varepsilon_t - 1, M) + 1$, $j = (\varepsilon_t - i^*)/M + 1$, $j^* = k_j$.
        if $s_{j^*} \leq \overline{S}_{i^*}$ then
            Set $x_{i^*j^*} = 1$, $\overline{S}_{i^*} \longleftarrow \overline{S}_{i^*} - s_{j^*}$, $\overline{C}_{i^*} \longleftarrow \overline{C}_{i^*} - a_{i^*j^*}$.
        end if
    end for
    Set $\lambda_{k_j} = 0$ for $\forall j$, $\mathcal{J} = \emptyset$.
end while
Calculate $z$ based on (8a).
Output: $z$, $\mathbf{X}$, $\mathbf{y}$.
```

given R_j $(\forall j)$, the random values r_{ij} $(\forall i, \forall j)$ should satisfy $\sum_{i=1}^M r_{ij} = R_j$ by setting $r_{ij} \leftarrow \frac{r_{ij}}{\sum_{i=1}^M r_{ij}} \cdot R_j$. All the simulation results are averaged over 100 random realizations and all codes are written in MATLAB (8.6.0) running on a personal computer (Intel Core i7-2600 CPU @3.40GHz, 8.00 GB RAM).

A. Small-scale Contents

Tables II and III compare the performance of the optimal BNB method and the proposed method as well as the lower bound in terms of computation time and objective value for small-scale contents. Here, $M = 5$, $S = 50$ Gbits and $C = 50$ Mbps. From Table II, as the number of contents goes up, the optimal BNB method takes exponentially increasing CPU time while the proposed method has a much lower time complexity. From Table III, the achieved optimal objective value with BNB method is very close to the lower bound since most elements of the solution $(\mathbf{X}_R^*, \mathbf{Y}_R^*)$ are binary, while the proposed method can also achieve suboptimal or even near-optimal objective value, though the gap between the objective values achieved by the BNB method and the proposed method gradually goes up with the number of contents. Thus, though the proposed method has some loss in performance relative to the optimal solution, it only takes a little time to find a suboptimal or even near-optimal feasible solution, which indicates that the proposed method has a good performance in both time complexity and solution search.

TABLE II
AVERAGE CPU COMPUTATION TIME (UNIT: SECOND)

# Content	100	200	300	400	500
BNB	0.827	47.063	142.930	818.141	2,410.397
Proposed	0.048	0.060	0.075	0.102	0.127

TABLE III
AVERAGE OBJECTIVE VALUE

# Content	100	200	300	400	500
LB	-2,230.7	-4,319.8	-6,314.5	-8,156.4	-9,816.7
BNB	-2,230.4	-4,319.3	-6,313.7	-8,155.0	-9,816.1
Proposed	-2,213.6	-3,871.3	-5,246.8	-6,266.0	-7,377.1

B. Large-scale Contents

As the time complexity of the optimal solution and the proposed method's good performance have been verified in the case of small-scale contents, we only consider the proposed method and choose the non-caching strategy and lower bound as baselines in the comparative evaluations for large-scale contents.

Fig. 2 compares the performance of different strategies in terms of the weighted expected sum of traffic loads versus number of contents with $M = 10$ PBSs and $S = 500$ Gbits. From Fig. 2, all the weighted expected sum of traffic loads goes up with the number of contents. The proposed caching strategies can greatly reduce the weighted expected sum of traffic loads and are close to the lower bound, especially when the maximum backhaul link rate C is relatively large.

Fig. 3 compares the performance of different strategies in terms of weighted expected sum of traffic loads versus total cache size (percentage to total content size) with $M = 10$ PBSs and $N = 5,000$ contents. From Fig. 3, all the weighted

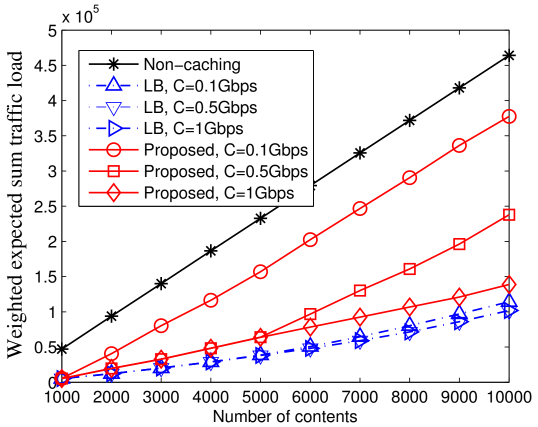

Fig. 2. Weighted expected sum of traffic load versus number of contents with $M = 10$ PBSs and $S = 500$ Gbits.

expected sums of traffic loads achieved by the proposed method and LB decrease significantly as the cache size increases while the weighted expected sum of traffic loads with non-caching method stays constant. Besides, for a fixed cache size, if the maximum backhaul link rate C is large enough, simply increasing C cannot decrease the weighted expected sum of traffic loads greatly since the rate constraints in (6c) are relaxed enough while the cache size constraints in (6b) is very tight.

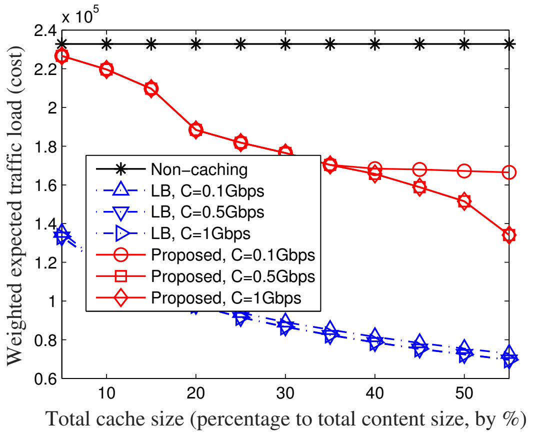

Fig. 3. Weighted expected sum of traffic load versus total cache size (percentage to total content size) with $M = 10$ PBSs and $N = 5,000$ contents.

V. Conclusion

In this paper, we have proposed caching strategies in HetNets to minimize the weighted expected sum of traffic loads of accessing the demanded contents under the constraints of PBSs' cache sizes and the backhaul link rates between PBSs and the MBS. We have transformed the formulated irregular optimization problem into a BILP problem and then proposed a low-complexity iterative greedy heuristic method. Evaluation results have shown that the proposed caching strategies can greatly decrease the network weighted expected sum of traffic loads and outperform non-caching strategies. As well, compared with the optimal exponential-time BNB method, the proposed method can get suboptimal or even near-optimal solution with much less time complexity. More studies on how PBS caching can deal with user mobility will be carried out in our future work.

Acknowledgment

This work is support in part by a China Scholarship Council Four Year Doctoral Fellowship, the Canadian Natural Sciences and Engineering Research Council through grants RGPIN-2014-06119 and RGPAS-462031-2014 and the National Natural Science Foundation of China through Grant No. 61271182.

References

[1] X. Wang, X. Li, V.C.M. Leung, and P. Nasiopoulos, "A Framework of Cooperative Cell Caching for the Future Mobile Networks," *in Proc. HICSS*, pp. 5404-5413, January 2015.

[2] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015," CISCO, Tech. Rep., 2011.

[3] X. Li, X. Wang, S. Xiao, and V.C.M. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks," *in Proc. IEEE ICC*, pp. 5652-5657, June 2015.

[4] X. Li, X. Wang, C. Zhu, W. Cai, and V.C.M. Leung, "Caching-as-a-Service: virtual caching framework in the cloud-based mobile networks," *in Proc. IEEE INFOCOM, Computer Communications Workshops*, pp. 372-377, May 2015.

[5] X. Wang, et al., "Cache In The Air: Exploiting Content Caching and Delivery Techniques for 5G Systems," *IEEE Communication Magazine*, vol. 52, no. 2, pp. 131-139, February 2014.

[6] K. Samdanis, T. Taleb, and S. Schmid, "Traffic Offload Enhancements for eUTRAN," *IEEE Communication Surveys & Tutorials*, vol. 14, no. 3, pp. 884-896, 2012.

[7] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," *In Usenix/ACM SIGCOMM IMC*, October 2007.

[8] H. ElSawy, E. Hossain, and M. Haenggi, "Stochastic Geometry for Modeling, Analysis, and Design of Multi-Tier and Cognitive Cellular Wireless Networks: A Survey," *IEEE Commun. Surveys and Tutorials*, vol. 15, no. 3, pp. 9961019, June 2013.

[9] C. Yang, et al., "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. PP, no. 99, 2015.

[10] S. Woo, E. Jeong, S. Park, J. Lee, S. Ihm and K. Park, "Comparison of Caching Strategies in Modern Cellular Backhaul Networks," *in ACM MobiSys*, pp. 319-332, 2013.

[11] N. Golrezaei et al., "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers" *in Proc. IEEE INFOCOM*, pp. 1107-1115, March 2012.

[12] B. Han, et al., "AMVS-NDN: Adaptive Mobile Video Streaming and Sharing in Wirelss Named Dada Networking," *in IEEE INFOCOM, NOMEN Workshop*, April 2013.

[13] H. Ahlehage and S. Dey, "Video Caching in Radio Access Network: Impact on Delay and Capacity," *in IEEE WCNC*, April 2013.

[14] V. Jain, and I.E. Grossmann, "Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems," *INFORMS Journal on Computing*, vol. 13, no. 4, pp. 258-276, 2001.

[15] S. Martello, and P. Toth. Knapsack Problems: Algorithms and Computer Implementations. John & Sons, Inc. New York, USA, 1990.