Provisioning Cost-Effective Mobile Video Caching

Seyed Ehsan Ghoreishi, Vasilis Friderikos, Dmytro Karamshuk, Nishanth Sastry, and A. Hamid Aghvami

Centre for Telecommunications Research, King's College London, London WC2R 2LS, U.K.
E-mail: seyed_ehsan.ghoreishi@kcl.ac.uk

***Abstract*—The exploding volumes of mobile video traffic call for deploying content caches inside mobile operators network. With in-network caching, users' requests for popular content can be served from a content cache deployed at mobile gateways in vicinity to the end user, therefore considerably reducing the load on the content servers and the backbone of operator's network. In practice, content caches can be installed at multiple levels inside an operator's network (e.g., serving gateway, packet data network gateway, RAN, etc.), leading to an idea of *hierarchical in-network video caching*. In order to evaluate the pros and cons of hierarchical caching, in this paper we formulate a cache provisioning problem which aims to find the best trade-off between the cost of cache storage and bandwidth savings from hierarchical caching. More specifically, we aim to find the optimal size of video caches at different layers of a hierarchical in-network caching architecture which minimizes the ratio of transmission bandwidth cost to storage cost. We overcome the complexity of our problem which is formulated as a binary-integer programming (BIP) by using canonical duality theory (CDT). Numerical results obtained using the invasive weed optimization (IWO) show that important gains can be achieved, with benefit-cost ratio and cost efficiency improvements of more than 43% and 38%, respectively.**

***Index Terms*—Cache storage, canonical duality, hierarchical in-network caching, invasive weed optimization, mobile video delivery.**

I. Introduction

The extensive growth in adoption of smartphones and tablets has led to a continuous increase in mobile video traffic. According to the recent reports [1], mobile video will represent 72% of global mobile data traffic by 2019, a 13-fold increase from 2014. This new phenomenon has urged mobile operators to redesign their networks and search for cost-effective solutions to bring content closer to the end user [2], [3].

One approach to this problem lies in installing geographically distributed content delivery networks (CDNs), which can efficiently serve users within certain geographic areas. However, in order to reach an end-user's device, CDN-served traffic must still traverse through the wireless carrier core network (CN) and radio access network (RAN) both of which contribute to delays in streaming video content. In contrast, with *in-network caching*, users can access popular content from caches of nearby mobile network operator (MNO) gateways [i.e. evolved packet core (EPC) and RAN] [3]–[8], therefore significantly reducing video streaming latency. Moreover, from the Internet Service Providers' (ISP) perspective, in-network caching also helps to reduce inter- and intra-ISP traffic and, so, to optimize operating costs for leasing expensive fiber lines that connect eNodeBs to EPC [7], [8].

Several approaches have been proposed to analyze intelligent caching strategies for mobile content caching inside MNO's network [5]–[7]. An extensive overview of the techniques for in-network content caching in 5G mobile networks has been introduced in [8], whereas different proactive mobile caching schemes have been discussed in [4], [9]. The current paper contributes to this stream of work by analyzing the trade-off between the potential savings from- and infrastructural costs of hierarchical in-network caching. In more details, the main contributions of this paper can be summarized as follows:

- To the best of our knowledge, we present the first attempt to formalize the problem of storage provisioning for a hierarchical in-network video caching which optimizes the trade-off between the cost of transmission bandwidth and the cost of storage.
- We focus our analysis on Scalable Video Coding (SVC)-based dynamic adaptive streaming over HTTP (DASH) format which encodes a video into different quality layers and is therefore more resource-efficient than traditional H.264/AVC-based DASH in which a separate AVC video file is encoded for each video quality format [10].
- We solve the storage provisioning problem using CDT [11]. More specifically, we transform our BIP problem into a canonical dual problem in continuous space, which is a concave maximization problem. Additionally, we provide the conditions under which the solutions of the canonical dual problem and primal problem are identical.
- The canonical dual problem results in complex non-linear equations which are efficiently solved by applying IWO algorithm [12].

In summary, our results suggest an improvement of up to 43.74% in benefit-cost ratio and 38.59% in cost efficiency in comparison with a naive Least Frequently Used (LFU) approach.

The rest of the paper is structured as follows. Section II describes the system model. The cache provisioning problem is formulated in III. Section IV presents the canonical dual framework. Section V conducts a simulation analysis of the model. The conclusion is presented in Section VI.

II. System Model

The system consists of I video streams, which are indexed by the set $\mathcal{I} \triangleq \{1, \ldots, i, \ldots, I\}$. We index different quality layers of a video stream by the set $\mathcal{J} \triangleq \{1, \ldots, j, \ldots, J\}$. By q_{ij}, we denote the j^{th} quality layer of video i, which

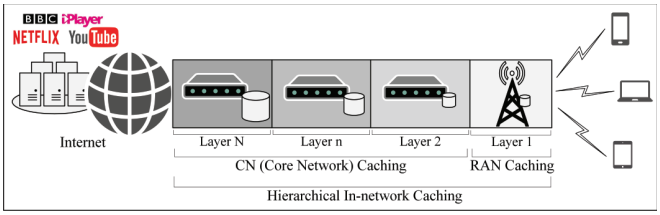

Fig. 1: A hierarchical in-network video caching system.

has a size and popularity (hit rate) of f_{ij} and p_{ij}, respectively. We consider a hierarchical in-network caching system with caches within different layers as shown in Fig. 1. We index different layers of the hierarchical architecture by $\mathcal{N} \triangleq \{1, \ldots, n, \ldots, N\}$. One example of a hierarchical in-network caching system can be found in [5], which defines a cache hierarchy tree of three layers with first, second and third layer nodes being eNodeBs, serving gateways (S-GWs) and packet data network gateway (P-GW), respectively. More examples can be found in [7].

A. Notations and Variables

1) Cache Assignment Binary Decision Variable ($x_{nij} \in \{0,1\}$): represents the cache assignment for q_{ij} in the n^{th} cache hierarchy, where $x_{nij} = 1$ indicates that an storage size of f_{ij} should be assigned to a cache in layer n of the hierarchical in-network caching system and $x_{nij} = 0$ otherwise.

2) Provisioned Storage Size (s_n): the storage capacity that is required to be assigned to the n^{th} layer of the in-network hierarchy.

3) Maximum Possible Storage Size (m_n): the maximum possible storage capacity that the MNO can install on the n^{th} layer of hierarchical caching system.

4) Effective Load (l_{nij}): is the reduction in the transmission bandwidth as a result of caching q_{ij} in layer n of the in-network caching hierarchy, where $l_{nij} = f_{ij} \times p_{ij} \times x_{nij}$.

5) Benefit Function (b_n): We assume that the benefit of transmission bandwidth saving follows a predefined function $\Gamma : \mathbb{R} \rightarrow \mathbb{R}$. Thus, we estimate the benefit derived from the reduction in transmission bandwidth when videos are cached in the n^{th} layer of the in-network caching hierarchy as

$$b_n(l_{nij}) = \Gamma\Big(\sum_{i=1}^{I}\sum_{j=1}^{J} l_{nij}\Big) \qquad \forall n \in \mathcal{N}. \tag{1}$$

6) Cost Function (c_n): We assume that the cache storage cost follows a predefined function $\Lambda : \mathbb{R} \rightarrow \mathbb{R}$. Hence, the cost associated with provisioned storage size s_n is

$$c_n(s_n) = \Lambda(s_n) \qquad \forall n \in \mathcal{N}. \tag{2}$$

Both benefit and cost functions can be any appropriate function defined by the MNO. However, without loss of generality, we may assume that they are either linear or logarithmic.

III. Problem Formulation

We formulate the cache provisioning problem as follows.

$$\max_{\mathbf{x}} \frac{\sum_{n=1}^{N} b_n(l_{nij})}{\sum_{n=1}^{N} c_n(s_n)} \tag{3}$$

subject to:

$$s_n = \sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij} x_{nij} \leq m_n \qquad \forall n \in \mathcal{N} \tag{3a}$$

$$\sum_{n=1}^{N} x_{nij} \leq 1 \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{3b}$$

$$x_{nij-1} \geq x_{nij} \qquad \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}-\{1\} \tag{3c}$$

$$x_{nij} \in \{0,1\} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N}. \tag{3d}$$

The objective of optimization problem (3) is to find the optimal provisioned storage capacity, s_n, which maximizes the ratio of overall benefit (1) to overall cost (2). Constraint (3a) ensures that the cache storage allocated to the n^{th} layer of the hierarchical caching system is upper-bounded by the maximum possible storage capacity threshold, m_n. Constraint (3b) indicates that each video can be cached in one hierarchical layer inside the in-network caching architecture exclusively. Constraint (3c) ensures that if a video quality layer is cached, all the lower quality layers are cached too. We use binary variables $x_{nij} \in \{0,1\}$ explained in section II-A1.

The optimization problem (3) is difficult to solve due to its combinatorial nature. As an intermediate step towards solution, we convert (3) into a BIP problem by defining a cache allocation matrix where instead of making decisions on the basis of individual video quality layer, decisions are made on the basis of feasible set of video layer cache allocation patterns that satisfies constraint (3c). The idea of pattern allocation is similar to [13]. We index all the combinations of video streams and the respective quality layers by the set $\mathcal{K} \triangleq \{1, \ldots, k, \ldots, K\}$. $K = |\mathcal{K}|$ denotes the cardinality of set $\mathcal{K}$. The cache allocation matrix is of the order $K \times A$, where each row corresponds to the video stream-video quality layer combination index and each column corresponds to a feasible cache allocation pattern [meeting constrain (3c)]. We denote by A the total number of feasible allocation patterns. The basic idea of this cache allocation matrix, for the case of 2 videos each with 2 quality layers is illustrated by (4). In any allocation pattern (i.e., any column), a "1" is placed when the video quality layer is cached, otherwise a "0" is placed.

$$\mathbf{Y}^n = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{4}$$

We define a cache indicator vector $\mathbf{x} \triangleq [\mathbf{x}_n]_{N\times 1}$, where $\mathbf{x}_n = [x_{na}]_{A\times 1}$. Each entry $x_{na} \in \{0,1\}$ indicates whether the cache allocation pattern a is allocated to hierarchical caching layer n or not.

Note that all the caches in the hierarchical in-network caching system have the same allocation patterns matrix. We rewrite (3) as a BIP problem as follows:

$$\min_{\mathbf{x}} \left\{ \mathcal{P}(\mathbf{x}) = -\frac{\sum_{n=1}^{N}\sum_{a=1}^{A} b_{na}x_{na}}{\sum_{n=1}^{N}\sum_{a=1}^{A} c_{na}x_{na}} \right\} \tag{5}$$

subject to:

$$\sum_{a=1}^{A} s_{na}x_{na} \leq m_n \qquad \forall n \in \mathcal{N} \quad (5a)$$

$$\sum_{n=1}^{N}\sum_{a=1}^{A} Y_{ka}^{n}x_{na} \leq 1 \qquad \forall k \quad (5b)$$

$$x_{na}\left(x_{na}-1\right) = 0 \qquad \forall n \in \mathcal{N}, \forall a \quad (5c)$$

$$\sum_{a=1}^{A} x_{na} = 1 \qquad \forall n \in \mathcal{N}. \quad (5d)$$

where b_{na} and c_{na} are the transmission bandwidth benefit and storage cost of allocating pattern a to hierarchical cache layer n, which results in a provisioned storage size of s_{na}. For a cache layer n, constraint (5a) puts an upper-bound of m_n on the provisioned storage size, which is equivalent to constraint (3a). Constraint (5b) ensures the exclusivity of the allocated videos, where Y_{ka}^{n} denotes the k^{th} row and a^{th} column of the matrix $\mathbf{Y}^n$, where $Y_{ka}^{n} = 1$ indicates that the video stream-video quality layer combination k should be cached in hierarchical layer n and $Y_{ka}^{n} = 0$ otherwise. Constraint (5c) is a pure binary constraint that ensures $x_{na} \in \{0,1\}$. Constraint (5d) ensures that at most one allocation pattern is chosen for each caching layer.

Although the optimization problem (5) is simpler and more tractable than (3), the solution is still exponentially complex.

IV. Canonical Dual Framework

A. Dual Problem Formulation

We convert our BIP problem (5) into a continuous space canonical dual problem using CDT [11], [14], which is solved in continuous space. We then identify the conditions under which the solution of the canonical dual problem is identical to that of the primal. A generic framework for solving 0-1 quadratic problems using CDT can be found in [15]. However, due to additional constraints, our problem is more complex. A framework for solving resource allocation BIP problems using CDT is given in [16], which will be extended to solve (5).

We define the feasible space for the primal problem (5) by $\mathcal{X}_p = \left\{\mathbf{x} \in \{0,1\}^{NA}\right\}$. We temporarily relax the equality constraints (5c) and (5d) to inequalities and transform the primal problem with these inequality constraints into continuous domain canonical dual problem. We then solve the problem in continuous space and provide the conditions under which the solutions of the canonical dual problem and primal problem are identical.

As a key step towards canonical dual formulation, we define the geometrical operator for the primal problem as $\wedge(\mathbf{y}) = (\boldsymbol{\delta}, \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\sigma}) \in \mathcal{Y}_g$, which is a vector valued mapping where $\mathcal{Y}_g$ is the feasible space for $\mathbf{y}$, and

$$\begin{cases} \boldsymbol{\lambda} = [\sum_{a=1}^{A} s_{na}x_{na} - m_n]_{N\times 1} \\ \boldsymbol{\mu} = [\sum_{n=1}^{N}\sum_{a=1}^{A} Y_{ka}^{n}x_{na} - 1]_{K\times 1} \\ \boldsymbol{\nu} = [x_{na}\left(x_{na}-1\right)]_{NA\times 1} \\ \boldsymbol{\tau} = [\sum_{a=1}^{A} x_{na} - 1]_{N\times 1} \end{cases} \tag{6}$$

Therefore, the feasible space for $\mathbf{y}$ is defined by $\mathcal{Y}_g = \mathbb{R}^N \times \mathbb{R}^K \times \mathbb{R}^{NA} \times R^N | \boldsymbol{\lambda} \leq 0, \boldsymbol{\mu} \leq 0, \boldsymbol{\nu} \leq 0, \boldsymbol{\tau} \leq 0$.

Next, we define the indicator function [15] as

$$V(\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \leq 0 \\ +\infty & \text{otherwise.} \end{cases} \tag{7}$$

We rewrite the primal problem (5) in the canonical form using indicator function (7) as follows:

$$\min\left\{V(\wedge(\mathbf{y})) + \mathcal{P}(\mathbf{x})\right\}. \tag{8}$$

We now define $\mathbf{y}^* = (\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^*)$ as the vector of dual variables associated with the corresponding restrictions $\mathbf{y} \leq 0$. The feasible space for $\mathbf{y}^*$ is defined by $\mathcal{Y}_d = \mathbb{R}^N \times \mathbb{R}^K \times \mathbb{R}^{NA} \times R^N | \boldsymbol{\lambda}^* \geq 0, \boldsymbol{\mu}^* \geq 0, \boldsymbol{\nu}^* \geq 0, \boldsymbol{\tau}^* \geq 0$. Based on the Fechnel transformation, the canonical sup-conjugate function of $V(\mathbf{y})$ is defined as

$$\begin{aligned} V^*(\mathbf{y}^*) &= \sup\left\{\langle \mathbf{y}, \mathbf{y}^*\rangle - V(\mathbf{y}) | \mathbf{y} \in \mathcal{Y}_g, \mathbf{y}^* \in \mathcal{Y}_d\right\} \\ &= \sup_{\mathbf{y}^*}\left\{\langle \boldsymbol{\lambda}^T\boldsymbol{\lambda}^* + \boldsymbol{\mu}^T\boldsymbol{\mu}^* + \boldsymbol{\nu}^T\boldsymbol{\nu}^* + \boldsymbol{\tau}^T\boldsymbol{\tau}^* - \mathcal{Y}_g\rangle\right\} \\ &= \begin{cases} 0 & \text{if } \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^* \geq 0 \\ +\infty & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

Using the definition of sub-differential, it can be easily verified that if $\mathbf{y}^* > 0$, then the condition $\mathbf{y}^T\mathbf{y}^* = 0$ leads to $\mathbf{y} = 0$, and consequently $\mathbf{x} \in \mathcal{X}_p$. Hence, the dual feasible space for the primal problem in (5) is an open positive cone defined by $\mathcal{X}_p^\sharp = \{\mathbf{y}^* \in \mathcal{Y}_d | \mathbf{y}^* > 0\}$.

We define the total complementarity function [11] as

$$\Xi(\mathbf{x}, \mathbf{y}^*) = \wedge(\mathbf{x})^T\mathbf{y}^* - V^*(\mathbf{y}^*) + \mathcal{P}(\mathbf{x}), \tag{10}$$

which is obtained by replacing $V(\mathbf{y}) = \wedge(\mathbf{x})^T\mathbf{y}^* - V^*(\mathbf{y}^*)$ (Fechnel-Young equality) in (8). We use the definitions of $\wedge(\mathbf{x}), V^*(\mathbf{y}^*)$ and $\mathcal{P}(\mathbf{x})$ to express $\Xi(\mathbf{x}, \mathbf{y}^*) = \Xi(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^*)$ as given by

$$\begin{aligned} \Xi(\mathbf{x}, \mathbf{y}^*) = &\sum_{n=1}^{N}\sum_{a=1}^{A} x_{na}\Phi - \frac{\sum_{n=1}^{N}\sum_{a=1}^{A} b_{na}x_{na}}{\sum_{n=1}^{N}\sum_{a=1}^{A} c_{na}x_{na}} \\ &- \sum_{n=1}^{N}\lambda_n^* m_n - \sum_{k=1}^{K}\mu_k^* - \sum_{n=1}^{N}\tau_n^* + \sum_{n=1}^{N}\sum_{a=1}^{A}\nu_{na}^* x_{na}^2, \end{aligned} \tag{11}$$

where $\Phi = \sum_{k=1}^{K}\mu_k^* Y_{ka}^n + \lambda_n^* s_{na} + \tau_n^* - \nu_{na}^*$. Next, we define the canonical dual function [11], [15] using the canonical dual variables as

$$\Upsilon(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^*) = \text{sta}\left\{\Xi(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^*)\right\}, \tag{12}$$

where sta$(\cdot)$ denotes finding the stationary point of the function. We are primarily interested in the cache allocation vector $\mathbf{x}$ for a node n. The stationary point of $\Xi(\mathbf{x}, \mathbf{y}^*)$ occurs at

$$x_{na}(\mathbf{y}^*) = \frac{1}{2} - \frac{1}{2\nu^*_{na}} \left(\sum_{k=1}^{K} \mu^*_k Y^n_{ka} + \lambda^*_n s_{na} + \tau^*_n \right) \qquad \forall n, a, \tag{13}$$

where the stationary point is obtained through $\nabla_{\mathbf{x}} \Xi(\mathbf{x}, \mathbf{y}^*) = 0$. Using (12) and (13), we obtain the dual function, which is given by (14), shown at the next page.

The dual function is a concave function on $\mathcal{X}^\sharp_p$. The canonical dual problem associated with (5) can be formulated as

$$\min \{\mathcal{P}(\mathbf{x}) | \mathcal{X}_p\} = \max \left\{ \Upsilon(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^*) | \mathcal{X}^\sharp_p \right\}. \tag{15}$$

Theorem 1. If $\mathcal{P}(\tilde{\mathbf{x}}) = \Upsilon(\tilde{\mathbf{y}}^*)$ where $\tilde{\mathbf{x}}$ denotes the KKT point of the primal problem and $\tilde{\mathbf{y}}^* = (\tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*, \tilde{\boldsymbol{\nu}}^*, \tilde{\boldsymbol{\tau}}^*) \in \mathcal{X}^\sharp_p$ denotes the KKT point of the dual function, there exists a perfect duality relationship between the primal problem in (5) and its canonical dual problem.

Proof. The proof directly extends from [14]. ■

Theorem 1 shows that the BIP in (5) is converted into a continuous space canonical dual problem which is perfectly dual to it. Moreover, the KKT point of the dual problem provides the KKT point of the primal problem.

Theorem 2. (global optimality conditions): If $\tilde{\mathbf{y}}^* = (\tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*, \tilde{\boldsymbol{\nu}}^*, \tilde{\boldsymbol{\tau}}^*) \in \mathcal{X}^\sharp_p$, then $\tilde{\mathbf{x}}$ is a global minimizer of $\mathcal{P}(\mathbf{x})$ over $\mathcal{X}_p$ and $\tilde{\mathbf{y}}^*$ is a global maximizer of $\Upsilon(\tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*, \tilde{\boldsymbol{\nu}}^*, \tilde{\boldsymbol{\tau}}^*)$ over $\mathcal{X}^\sharp_p$. Hence, $\mathcal{P}(\tilde{\mathbf{x}}) = \min \{\mathcal{P}(\mathbf{x}) | \mathcal{X}_p\} = \max \left\{ \Upsilon(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \boldsymbol{\tau}^*) | \mathcal{X}^\sharp_p \right\} = \Upsilon(\tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*, \tilde{\boldsymbol{\nu}}^*, \tilde{\boldsymbol{\tau}}^*)$.

Proof. The proof directly extends from [14]. ■

According to Theorem 2, if the given global optimality conditions are met, the solution of the canonical dual problem provides an optimal solution to the primal problem. Solving the KKT conditions associated with the dual function in (14) is necessary and sufficient for global optimality as the dual problem is a concave maximization problem over $\mathcal{X}^\sharp_p$.

The KKT conditions of the dual function in (14) are given by $(\partial\Upsilon/\partial\lambda^*_n) = 0, (\partial\Upsilon/\partial\mu^*_k) = 0, (\partial\Upsilon/\partial\nu^*_{na}) = 0$ and $(\partial\Upsilon/\partial\tau^*_n) = 0$, where the respective partial derivatives are given by (16)-(19), shown at the next page.

B. Invasive Weed Optimization Algorithm

Traditional gradient-based algorithms exist in literature for solving the non-linear equations resulting from the KKT conditions associated with the dual function. However, they show many defects such as oscillatory behavior, sensitivity to choice of initial values and complexity associated with the differentiation of KKT conditions and calculation of step size.

We deploy an IWO [12] algorithm for solving the complex non-linear equations associated with the KKT conditions [17]. Inspired by the invasive and robust nature of weeds, IWO is an evolutionary optimization algorithm, which has been shown to perform better than traditional approaches in terms of convergence. It also has the desirable properties of dealing with non-differentiable and complex objective functions and does not show the aforementioned defects.

In summary, the key steps of IWO are *Initialization*, where seeds are randomly dispersed over the search space; *Reproduction*, where every seed grows to a flowering plant and produces seeds; *Spatial Dispersion*, where produced seeds are distributed based on a normal distribution with a mean of zero and standard deviation reducing from an initial value σ_{initial} to a final value σ_{finala} according to equation $\sigma_{\text{iter}} = [(iter_{\max} - iter)/iter_{\max}]^g (\sigma_{\text{initial}} - \sigma_{\text{final}}) + \sigma_{\text{final}}$, where g is the modulation index; and *Competitive Exclusion*, where a competitive mechanism is implemented for eliminating undesirable plants. A detailed discussion on IWO is out of scope of this paper. Interested reader is referred to [12], [18].

Algorithm 1: Hierarchical caching based on IWO

```
initialize λ*, μ*, ν*, τ*, ∀n ∈ N, iter = 0;
∀ ∂ϒ/∂ν*, where ν* ∈ (δ*, μ*, ν*, τ*)
create initial population of Q individuals (weeds):
W = {W_1, ···, W_Q};
while |ν*| > ϱ or iter = iter_max do
    evaluate the fitness of each individual i.e., calculate
    f(W_n), ∀n ∈ W;
    sort W in ascending order according to f(W_n);
    select the first Q_p individuals of W to create the set
    W_p;
    ∀ W_j, j = 1, ···, Q_p
    generate
    S_j = (f(W_j) − f_worst)/(f_best − f_worst) × (S_max − S_min) + S_max seeds;
    create population of generated seeds, W_s = {W_s};
    for i = 1 : |W_s| do
        W_s^i ← W_s^i + φ^i, where φ^i ~ L(0, σ_iter);
    end
    create W* = W ∪ W_s;
    sort W* in ascending order according to fitness;
    select the first Q_max individuals of W* and create W;
end
select the best fitted individuals λ*, μ*, ν* and τ*;
calculate x_n using (13);
```

V. Simulation Results

We assume a hierarchical in-network caching system consisting of 4 layers. Without loss of generality, we initially set the maximum possible storage capacities of hierarchical caching layers 1, 2, 3 and 4 to 200, 400, 500 and 600 gigabytes, respectively. In order to analyze the effects of maximum possible storage capacity on the performance of our proposed approach, we extend the cache size in increments of 20% until the maximum storage capacity of the first, second, third and forth layer caches reach 600, 1200, 1500 and 1800 gigabytes (typical storage capacities available today). In defining the cost and benefit functions, we assume that caching in the lower layers of the in-network caching system is

$$\Upsilon(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{\nu}^*,\boldsymbol{\tau}^*) = -\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{\Phi^2}{4\nu_{na}^*} - \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}\Phi}{2\nu_{na}^*}}{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}} - \sum_{n=1}^{N}\lambda_n^* m_n - \sum_{k=1}^{K}\mu_k^* - \sum_{n=1}^{N}\tau_n^*. \tag{14}$$

$$\frac{\partial\Upsilon}{\partial\lambda_n^*} = -\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{s_{na}\Phi}{2\nu_{na}^*} - \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}s_{na}}{2\nu_{na}^*}}{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}} + \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}s_{na}}{2\nu_{na}^*}\cdot\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}\Phi}{2\nu_{na}^*}}{\left(\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}\right)^2} - \sum_{n=1}^{N}m_n, \tag{16}$$

$$\frac{\partial\Upsilon}{\partial\mu_k^*} = -\sum_{n=1}^{N}\sum_{a=1}^{A}\left(\frac{\sum_{k=1}^{K}Y_{ka}^n}{2\nu_{na}^*}\Phi\right) - \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}\sum_{k=1}^{K}Y_{ka}^n}{2\nu_{na}^*}}{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}} + \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\sum_{k=1}^{K}Y_{ka}^n}{2\nu_{na}^*}\cdot\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}\Phi}{2\nu_{na}^*}}{\left(\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}\right)^2} - K, \tag{17}$$

$$\begin{aligned}\frac{\partial\Upsilon}{\partial\nu_{na}^*} =& \sum_{n=1}^{N}\sum_{a=1}^{A}\left[\left(\frac{\Phi}{2\nu_{na}^*}\right)^2 + \left(\frac{\Phi}{2\nu_{na}^*}\right)\right] - \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\left(\frac{-b_{na}\Phi}{2{\nu_{na}^*}^2} - \frac{b_{na}}{2\nu_{na}^*}\right)\cdot\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}\Phi}{2\nu_{na}^*}}{\left(\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}\right)^2} \\ &+ \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\left(\frac{-c_{na}\Phi}{2{\nu_{na}^*}^2} - \frac{c_{na}}{2\nu_{na}^*}\right)}{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}},\end{aligned} \tag{18}$$

$$\frac{\partial\Upsilon}{\partial\tau_n^*} = -\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{\Phi}{2\nu_{na}^*} - \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}}{2\nu_{na}^*}}{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}} + \frac{\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}}{2\nu_{na}^*}\cdot\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{b_{na}\Phi}{2\nu_{na}^*}}{\left(\sum_{n=1}^{N}\sum_{a=1}^{A}\frac{c_{na}\Phi}{2\nu_{na}^*}\right)^2} - N. \tag{19}$$

more costly and results in more transmission bandwidth saving benefit. We consider the total number of popular videos to be 4000 with 3 popular quality layers. As in [19], [20], we assume the video popularity is Zipf-like with a parameter of 0.6 and the video file sizes follow a Pareto (0.25) distribution with a minimum size of 60 megabytes.

We solve the KKT conditions for each dual variable associated with the dual problem deploying IWO and compute the allocation vector $\mathbf{x}_k$ using (13). A pseudo code for the cache provisioning algorithm is given as Algorithm 1. TABLE I provides a summary of the simulation parameters for IWO.

Fig. 2 compares the effect of using a logarithmic function with a linear function in identifying the optimal provisioned storage size under maximum possible capacity varying from 1.7 to 5.1 terabytes (20% increments). In both scenarios, an increase in the storage capacity increases the identified provisioned cache size. We note that when a maximum possible capacity of approximately 3.7 terabytes is reached, the in-network caching system possesses most of the popular videos worthy of being cached. Therefore, further increasing the maximum storage capacity does not lead to a noticeable increase in the provisioned cache size at this point .

TABLE I: IWO Numerical Parameter Values

Parameter	**Value**
Size of initial population (Q)	20
Min. fitness threshold (ϱ)	10^{-7}
Max. no. of iterations ($iter_{\max}$)	500
Max. no. of plants ($Q_{\max}$)	10
No. of seeds ($S_{\max}, S_{\min}$)	(5,0)
Non-linear modulation index	2.5
Standard deviation ($\sigma_{\text{initial}}, \sigma_{\text{final}}$)	(10,0.01)

We compare our proposed approach with the case when no storage provisioning is performed within the hierarchical in-network caching system and popular contents are cached using least frequently used (LFU) caching algorithm [21]. LFU caches the most popular videos in the lower layer caches closer to the end users [22]. In contrast with the other widely used caching algorithm, least recently used (LRU), LFU focuses on historical popularity over a long period of time. As a cache provisioning technique, our approach also considers a long term content popularity. Therefore, it is pertinent to compare

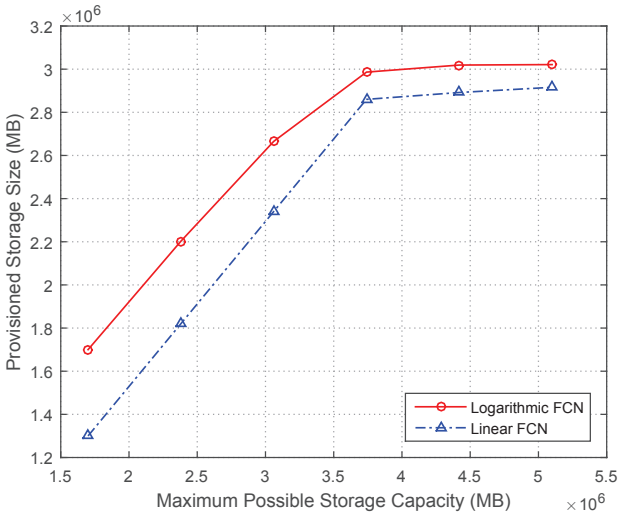

Fig. 2: Provisioned storage vs. maximum possible storage.

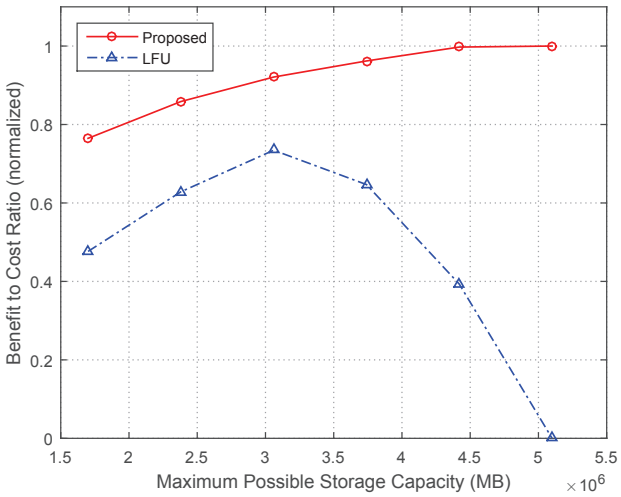

Fig. 3: Benefit-cost ratio vs. maximum possible storage.

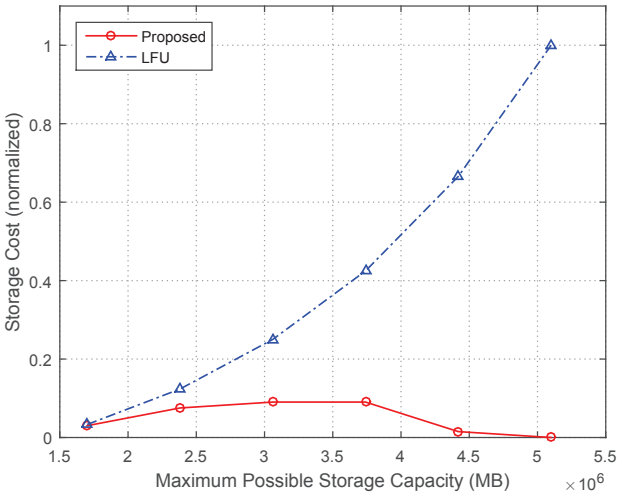

Fig. 4: Storage cost vs. maximum possible storage.

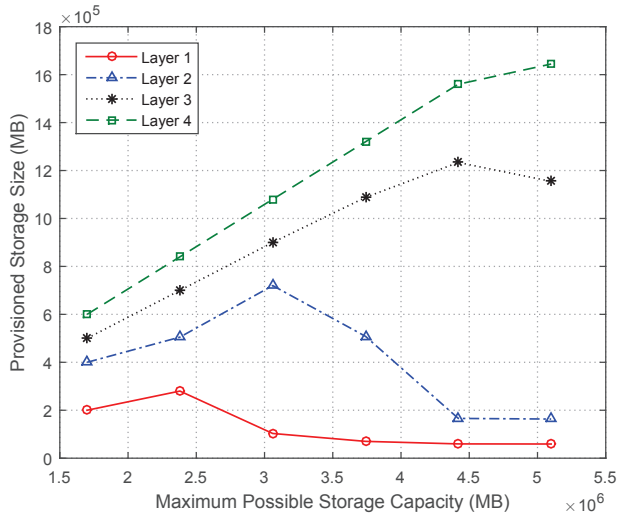

Fig. 5: Provisoned storage of different layers of hierarchical caching system vs. maximum possible storage.

our proposed scheme with LFU.

Fig. 3 compares the performance of our proposed approach with LFU in terms of benefit to cost ratio under different maximum possible storage capacities mentioned earlier. We note that our proposed approach improves benefit to cost ratio by 43.74%. When there is approximately 3.1 terabytes of storage capacity available, the benefit to cost ratio performance of LFU starts degrading as by this point, most of the popular videos have been cached and adding more storage only increases the cost for the same amount of saving in transmission bandwidth.

Fig. 4 compares the storage cost-effectiveness of our proposed approach with LFU. Since LFU does not support intelligent storage provisioning and uses the maximum storage capacity available, extending the cache size in increments of 20% increases the storage cost exponentially. However, our scheme only uses an optimal portion of the maximum possible storage and hence, decreases the cost significantly. Our proposed approach improves cost-effectiveness by 38.59%. When there is 3.7 terabytes of storage available, the cost starts decreasing in our approach as there is more storage available on cheaper caches at higher layers. Therefore, to increase cost-effectiveness, some of the videos that were previously cached at the expensive lower layer caches move to the higher layers.

Fig. 5 indicates how the increase in the maximum storage capacity affects the provisioned storage size of the caches at each hierarchical layer of the in-network caching system. As more storage is available on the cheaper devices in higher layers, more provisioned storage size is allocated to the higher layer devices due to greater cost efficiency.

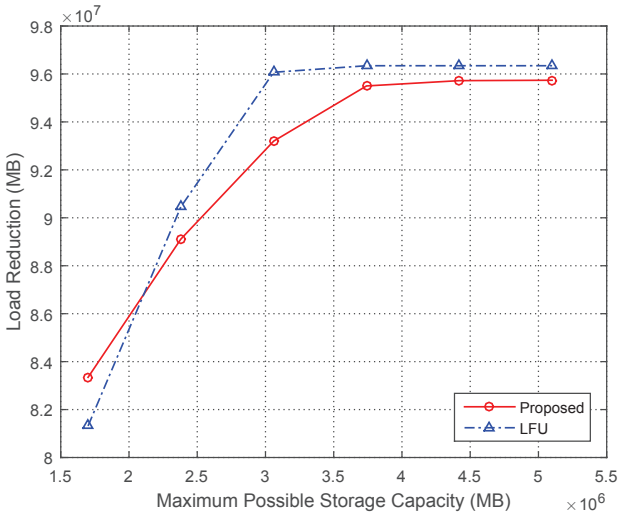

Fig. 6: Inter and intra-ISP traffic reduction vs. maximum possible storage.

Fig. 6 compares the reduction in inter and intra-Internet service provider (ISP) traffic as a result of deploying our proposed approach and LFU caching mechanism. It can be seen that LFU performs slightly better in terms of load reduction by only 0.764%, at the cost of considerably higher available storage, resulting in a significant increase in cost. It is worth noting that with LFU, upon availability of approximately 3.1 terabytes storage, most of the popular videos are cached and an increase in the maximum storage capacity does not further reduce the load in the CN.

Lastly, we discuss the complexity of the IWO algorithm. IWO is an iterative algorithm and is used for each dual variable associated with the dual function in (14). In each iteration for $\boldsymbol{\lambda}^* \geq 0, \boldsymbol{\mu}^* \geq 0, \boldsymbol{\nu}^* \geq 0, \boldsymbol{\tau}^* \geq 0$, we compute N, K, NA, and N variables, respectively. Therefore, it has an overall worst case complexity of $\mathcal{O}(iter_{\max} \cdot \{2N + K + NA\})$. A detailed performance evaluation of IWO algorithm in terms of convergence and computational time in compare with various algorithms such as Genetic Algorithm and Particle Swarm Optimization can be found in [12] and [18].

VI. Conclusion

In this paper, we have proposed a cache provisioning scheme which optimizes cache storage allocation inside a hierarchical in-network caching system in order to minimize both storage and transmission bandwidth costs. We use CDT to convert our BIP problem into its canonical dual. We use the IWO algorithm to obtain the solution of the dual problem. Numerical and simulation results have shown that the proposed scheme outperforms LFU algorithm by more than 43% and 38% in terms of benefit-cost ratio and cost-efficiency improvement, respectively.

Acknowledgment

This work is partially funded by the European Commission Framework Programme 7 Marie Curie Initial Training Networks (ITN) CROSSFIRE project (FP7-PEOPLE-317126).

References

[1] Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019*, Cisco White Paper, February 2015.

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

[3] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, August 2014.

[4] H. Abou-Zeid and H. Hassanein, "Toward green media delivery: Location-aware opportunities and approaches," *Wireless Commun.*, vol. 21, no. 4, pp. 38–46, 2014.

[5] H. Ahlehagh and S. Dey, "Hierarchical video caching in wireless cloud: Approaches and algorithms," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, June 2012.

[6] ——, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Trans. Netw.*, vol. 22, pp. 1444–1462, October 2014.

[7] S. Spagna, M. Liebsch, R. Baldessari, S. Niccolini, S. Schmid, R. Garroppo, K. Ozawa, and J. Awano, "Design principles of an operator-owned highly distributed content delivery network," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 132–140, 2013.

[8] 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 Commun. Mag.*, vol. 52, no. 2, pp. 131–139, 2014.

[9] H. Ahlehagh and S. Dey, "Adaptive bit rate capable video caching and scheduling," in *Proc. IEEE WCNC*. IEEE, April 2013, pp. 1357–1362.

[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sept 2007.

[11] D. Yang Gao, "Canonical dual transformation method and generalized triality theory in nonsmooth global optimization," *J. Global Optim.*, vol. 17, no. 1–4, pp. 127–160, 2000.

[12] A. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355–366, 2006.

[13] I. Wong, O. Oteri, and W. Mccoy, "Optimal resource allocation in uplink SC-FDMA systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2161–2165, May 2009.

[14] D. Yang Gao, N. Ruan, and H. D. Sherali, "Canonical dual solutions for fixed cost quadratic programs," in *Optimization and optimal control*. Springer, 2010, pp. 139–156.

[15] D. Y. Gao, R. lin Sheu, S. yi Wu, and C. K. L. Teo, "Canonical dual approach for solving 0-1 quadratic programming problems," *J. Ind. Manag. Optim.*, vol. 4, pp. 125–142, 2007.

[16] A. Ahmad and M. Assaad, "Polynomial-complexity optimal resource allocation framework for uplink systems," in *Proc. IEEE Global Telecoms. Conf. (GLOBECOM)*, 2011.

[17] A. Aijaz, M. Tshangini, M. Nakhai, X. Chu, and A.-H. Aghvami, "Energy-efficient uplink resource allocation in LTE networks with M2M/H2H co-existence under statistical QoS guarantees," *IEEE Trans. Commun.*, vol. 62, no. 7, pp. 2353–2365, July 2014.

[18] E. Pourjafari and H. Mojallali, "Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering," *Swarm and Evolutionary Computation*, vol. 4, pp. 33–43, 2012.

[19] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: A view from the edge," in *ACM SIGCOMM*, October 2007.

[20] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, October 2009.

[21] J. Wang, "A survey of web caching schemes for the internet," *ACM SIGCOMM Comput. Commun. Rev.*, October 1999.

[22] J. Ardelius, B. Grönvall, L. Westberg, and A. Arvidsson, "On the effects of caching in access aggregation networks," in *ACM ICN Workshop on Information-centric Networking*, 2012.