# A Cost-Driven Approach to Caching-as-a-Service in Cloud-Based 5G Mobile Networks

<table>
<thead>
<tr>
<th>Journal:</th>
<th>Transactions on Mobile Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID</td>
<td>TMC-2016-03-0187.R2</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Regular</td>
</tr>
<tr>
<td>Keywords:</td>
<td>H.5.1.g Video &lt; H.5.1 Multimedia Information Systems &lt; H.5 Information Interfaces and Representation (HCI) &lt; H Information Techn, G.1.6.b Convex programming &lt; G.1.6 Optimization &lt; G.1 Numerical Analysis &lt; G Mathematics of Computing, C.2.8.c Mobile communication systems &lt; C.2.8 Mobile Computing &lt; C.2 Communication/Networking and Information Technology &lt; C Comp</td>
</tr>
</tbody>
</table>
A Cost-Driven Approach to Caching-as-a-Service in Cloud-Based 5G Mobile Networks

Seyed Ehsan Ghoreishi, Member, IEEE, Dmytro Karamshuk, Member, IEEE, Vasilis Friderikos, Member, IEEE, Nishanth Sastry, Member, IEEE, Mischa Dohler, Fellow, IEEE, and A. Hamid Aghvami, Fellow, IEEE

Abstract—The exploding volumes of mobile video traffic call for deploying content caches inside mobile operator networks. 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. This inherently reduces the load on the content servers and the backbone of operator’s network. In light of the increasing trend in virtualization of network functions, we propose a cost-effective caching as a service (CaaS) framework for virtual video caching in 5G mobile networks. In order to evaluate the pros and cons of our CaaS approach, we formulate two virtual caching problems, namely maximum return on investment (MRI) and maximum offloaded traffic (MOT). MRI aims at maximizing return on caching investment by finding the best trade-off between the cost of cache storage and bandwidth savings from caching video contents in the mobile network operator (MNO)’s cloud. Likewise, MOT aims to maximize the traffic offloaded from the MNO’s core and backhaul within given budget constraints. More specifically, taking the popularity and size of video contents into account, MRI and MOT aim to find the optimal caching tables which maximize the ratio of transmission bandwidth cost to storage cost and the offloaded traffic for a given budget, respectively. We reduce the complexity of the proposed problem formulated as a binary-integer programming (BIP) by using canonical duality theory (CDT). Experimental results obtained using the invasive weed optimization (IWO) have shown significant performance enhancement of the proposed system in terms of return on investment, quality, offloaded traffic and storage efficiency.

Index Terms—Caching-as-a-Service (CaaS), virtual caching, mobile video delivery, canonical duality, invasive weed optimization, return on investment maximization.

1 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 mobile operator’s core network and access network (RAN). The significant strain on the operator’s core network and RAN backhaul contributes to congestion, delays in streaming video content and a constraint on the network’s capacity to serve a large number video requests currently. In contrast, with in-network caching, users can access popular content from caches of nearby MNO gateways, i.e. evolved packet core (EPC) and RAN [3], [4], [5], [6], therefore significantly reducing video streaming latency, congestion and increasing the capacity of the network to serve video content.

From the MNO’s perspective, in-network caching also helps to reduce inter- and intra-MNO traffic and optimize operating costs for leasing expensive fiber lines that connect eNodeBs to EPC [5], [6]. The reduction in the outbound traffic from the content provider’s users associated with the MNO decreases the traffic load directed to public CDN. This, in turn, inherently results in the content provider to pay less for CDN services.

Recently, the new trend of virtualizing mobile network functions into software-based cloud servers has been envisioned, which yields several advantages such as optimization of resource utilization, reduction in capital expenditures (CAPEX) and operating expenditures (OPEX), and increase in scalability and flexibility [7], [8]. The emergence of network function virtualization (NFV) [9] has stimulated research on the concepts of RAN as a service (RANaaS) and EPC as a service (EPCaaS). RANaaS virtualizes the traditional radio access processing functions into the cloud [10], with remote antennas [remote radio heads (RRHs)] connected with the servers running the virtualized baseband units (BBUs) in the MNO’s cloud center by high-speed fronthaul fiber networks. Likewise, with EPCaaS, some EPC network functions are instantiated on virtual machines (VMs) on top of a virtualized platform, running in a MNOs’ cloud centers [7].

The increasing drive towards the virtualization of mobile networks and services has motivated recent research aimed at proposing CaaS inside MNOs’ cloud centers [8]. With CaaS, rather than running traditional CDN services virtually, which are still statical storage of files and man-
agement units in VMs of the cloud, CaaS instances in the mobile cloud centers can be adaptively created, immigrated, scaled (up or down), shared and released depending on the user demands and requirements from third-party service providers and content providers. The MNO may charge content and service providers for caching their contents [11], whereas the operator guarantees a level of service for the cached contents.

In this paper, we propose a virtual caching policy in a cloud-based mobile operator network which maximizes the return on caching investment and offloaded traffic. To the best of our knowledge, this problem has not been investigated before in proactive off-line scenarios. Offloaded traffic signifies the traffic load that would be directed to public CDNs in the absence of caching in the operator’s network. By proactive caching, we mean that the caching decisions are made before the appearance of any request for any content. By off-line caching, we mean that the caching scheme knows the popularities of the contents (i.e. the number of requests made for each content) [11].

1.1 Related Work

Many studies have proposed CDNs for Internet content [12], as well as CDN services running in the cloud [13]. However, as explained earlier, caching at Internet CDNs do not address the problems of latency and capacity for video delivery in wireless networks.

Reference [14] investigates the effectiveness of caching least frequently used (LFU) as published by Hulu, caching using least recently used (LRU) policy, and a combination of the two using traces collected from a university campus. Like the Internet caching techniques, the above Internet video caching techniques do not address the problem of video capacity or delay in cellular networks.

Some studies have developed caching techniques for ad hoc networks [15], [16]. However, these techniques are not applicable to the problem of video caching and delivery in cellular networks.

Several approaches have been proposed to analyze intelligent caching strategies for mobile content caching inside MNO’s network [5]. An extensive overview of the techniques for in-network content caching in 5G mobile networks has been introduced in [6], whereas different proactive mobile caching schemes in base stations have been discussed in [2], [4], [17]. These works however, do not address the problem of caching in a cloud-based mobile network. Furthermore, these theoretical studies for in-network caching and caching content in base stations lack practical implementation consideration. For instance, the caching approach proposed in [2] needs the presence of additional helper nodes where videos are cached, and for users to have access to multiple helper nodes, both of which may be hard to satisfy.

Reference [8] represents the first attempt to develop a virtualized caching system inside MNO’s cloud center. The differences between our work and the work of [8] are fourfold: 1) the work in [8] only minimizes inter- and intra-MNO traffic load and does not take cost-efficiency and caching costs into account; 2) reference [8] does not take the scalable video coding (SVC) video requirements into consideration; 3) the constraints on the capacity of the fronthaul is not taken into consideration in [8]; 4) the virtual caching problem proposed in [8] is solved using a simplistic algorithm, which runs relatively fast, however, rarely achieves an optimal allocation [18].

The work in [19] focused on optimizing caching in heterogeneous networks with the aim to allow collaboration between the different networks so that to entail the optimal offloading between different networks under the assumption of different content requests. assume a fully distributed caching environment at the level of the base stations. The works in [20] and [21] are closely related. [20] provides architectural views on caching in emerging cloudified 5G networks whilst outlining a number of techniques related to virtualization and caching. [21] assumes a fully distributed caching environment at the level of base stations. Furthermore, the work in [22] links Information-centric-networking (ICN) techniques with caching especially tailored in the case where user mobility might impact the performance and content can be retrieved by other ICN routers hosting the content closer to the point of attachment of the user.

1.2 Contributions and Outline

Our contributions with respect to this paper are stated as follows:

- To the best of our knowledge, we present the first attempt to formalize a virtual caching framework to maximize the return on caching investment. Our budget-constrained approach maximizes the offloaded traffic while meeting the maximum budget threshold.
- By introducing a quality priority factor in our optimization problem, we assign a higher weight to contents with larger bit-rate, hence prioritizing the high bit-rate contents over the low bit-rate ones. This, in turn, results in great improvements in the end user’s quality of experience (QoE)\(^1\), as the streaming application observes higher throughput, less latency and smaller start up and buffering times, which are the key QoE differentiators [23], [24].

- We focus our analysis on SVC-based dynamic adaptive streaming over HTTP (DASH) format, which encodes a single video into different quality layers. Therefore, it is more resource-efficient than traditional H.264/AVC-based DASH in which a separate AVC video file is encoded for each video quality format [25].

- We solve the virtual caching problem using canonical duality theory (CDT) [26]. More specifically, we transform our binary integer programming (BIP) problem into a canonical dual problem in continuous space, which is a concave 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 invasive weed optimization (IWO) algorithm [27].

- Our results provide insight into the gains achieved from the perspective of the end user, content provider and MNOs.

1. ITU-T FG IPTV, Liaison Statement 50, Definition of Quality of Experience, 2007
In summary, our results suggest an improvement of more than 32% in return on investment, 21% in quality, 32% in offloaded traffic and 17% in storage efficiency in comparison to a na"ive LFU approach.

The rest of the paper is structured as follows. Section 2 describes the system model. The virtual caching framework is formulated in 3. Section 4 presents the canonical dual framework. Section 5 conducts a simulation analysis of the model. The conclusion is presented in Section 6.

Below is a summary of the abbreviations used throughout this paper.

**SUMMARY OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>application programming interface.</td>
</tr>
<tr>
<td>BBU</td>
<td>baseband unit.</td>
</tr>
<tr>
<td>BIP</td>
<td>binary-integer programming.</td>
</tr>
<tr>
<td>CaaS</td>
<td>caching as a service.</td>
</tr>
<tr>
<td>CAPEX</td>
<td>capital expenditures.</td>
</tr>
<tr>
<td>CDN</td>
<td>content delivery network.</td>
</tr>
<tr>
<td>CDT</td>
<td>canonical duality theory.</td>
</tr>
<tr>
<td>DASH</td>
<td>dynamic adaptive streaming over HTTP.</td>
</tr>
<tr>
<td>EPC</td>
<td>evolved packet core.</td>
</tr>
<tr>
<td>EPCaaS</td>
<td>EPC as a service.</td>
</tr>
<tr>
<td>IWO</td>
<td>invasive weed optimization.</td>
</tr>
<tr>
<td>LFU</td>
<td>least frequently used.</td>
</tr>
<tr>
<td>LRU</td>
<td>least recently used.</td>
</tr>
<tr>
<td>MNO</td>
<td>mobile network operator.</td>
</tr>
<tr>
<td>MOT</td>
<td>maximum offloaded traffic.</td>
</tr>
<tr>
<td>MRI</td>
<td>maximum return on investment.</td>
</tr>
<tr>
<td>NFV</td>
<td>network function virtualization.</td>
</tr>
<tr>
<td>OPEX</td>
<td>operating expenditures.</td>
</tr>
<tr>
<td>QoE</td>
<td>quality of experience.</td>
</tr>
<tr>
<td>RAN</td>
<td>radio access network.</td>
</tr>
<tr>
<td>RANaaS</td>
<td>RAN as a service.</td>
</tr>
<tr>
<td>RRH</td>
<td>remote radio head.</td>
</tr>
<tr>
<td>RTT</td>
<td>round trip time.</td>
</tr>
<tr>
<td>SLA</td>
<td>service level agreement.</td>
</tr>
<tr>
<td>SVC</td>
<td>scalable video coding.</td>
</tr>
<tr>
<td>VM</td>
<td>virtual machine.</td>
</tr>
</tbody>
</table>

**2 SYSTEM MODEL**

We consider a virtual caching system inside the MNO’s infrastructure as shown in Fig. 1. If a content is not available in the MNO’s virtual cache, it needs to traverse the MNO’s core and virtual BBU pool to get to the RRHs in a cluster, from which it is transmitted to the end users. Likewise, in order to cache a content in the operator’s network, it needs to travel through the MNO core to be cached in the BBU pool, from which it is sent to the RRHs to be transmitted to the end users. The requests for the content are then served from the BBU pool in the MNO’s infrastructure. Each 3rd-party service provider and content provider can program with the virtual caching by CaaS application programming interfaces (APIs). A service level agreement (SLA) is defined between the MNO and content providers, which determines the MNO’s liabilities in providing the required resources to guarantee a level of service for the videos that have been cached. The MNO can dynamically charge for the resource utilization of the service and content providers.

The system consists of $I$ video streams, which are indexed by the set $J \triangleq \{1, \ldots, i, \ldots, I\}$. We index different quality layers of a video stream by the set $\tilde{J} \triangleq \{1, \ldots, j, \ldots, J\}$. By $q_{ij}$, we denote the $j^{th}$ quality layer of video object $i$, which has a size, source bit-rate and popularity (hit rate) of $f_{ij}$, $b_{ij}$ and $p_{ij}$, respectively. We index different clusters by $N \triangleq \{1, \ldots, n, \ldots, N\}$. One example of a cloud-based caching system architecture can be found in [8]. We use the following notations and variables:

**Cache Assignment Binary Decision Variable** ($x_{nij}$) represents an entry in the caching table $x$. $x_{nij} = 1$ indicates that content $q_{ij}$ is cached to serve users in cell $n$ while meeting the SLA on users’ experience of the content. If $x_{nij} = 0$ but content $q_{ij}$ is available in the cache ($\sum_{n=1}^{N} x_{nij} \geq 1$) to serve users in a cell $n'$ under SLA guarantees, requests for content $q_{ij}$ from users in cell $n$ can be directed to the cache without any SLA liabilities. If $\sum_{n=1}^{N} x_{nij} = 0$, requests for content $q_{ij}$ are routed to the root.

**Offloaded Traffic** ($l_{ij}$) is the traffic load that would be directed to public CDNs in the absence of virtual caching in the MNO’s network. In other words, it is the reduction in the transmission bandwidth as a result of caching $q_{ij}$, where $l_{ij} = f_{ij} \cdot p_{ij}$. We denote by $L_{n}$ the cached traffic for each cluster $n$, which is given by $L_{n} = \sum_{i=1}^{I} \sum_{j=1}^{J} l_{ij} \cdot x_{nij}$ for $n \in N$.

**Storage Size** ($S_{n}$) is the storage capacity allocated to cluster $n$ for cloud-based caching. For pricing purposes, we calculate the required storage under the assumption that cached files are not shared between different clusters. The total storage required for an individual cluster $n$ is $S_{n} = \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \cdot x_{nij}$ for $n \in N$. We use the binary decision variable $y_{ij}$ to find the total physical storage required $S = \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \cdot y_{ij}$, where $y_{ij} \forall i, j$ is given by

$$y_{ij} = \begin{cases} 1 & \text{if } \sum_{n=1}^{N} x_{nij} \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

Fig. 1. Cloud-based virtual caching architecture.
Fronthaul Capacity \( (B_{\text{max}}) \) is the bandwidth capacity of the link between the operator’s cloud center and cluster \( n \). It should be noted that in order to meet the SLA with content providers, the MNO needs to provision for the peak rather than average bandwidth.

**Quality Priority Factor** \( (\Omega) \) prioritizes the video contents with higher bit-rates over low bit-rate videos. The offered throughput under TCP is inversely proportional to connections round trip time (RTT) \[28\]. As shown in [4], in comparison with fetching data from public CDNs, caching contents inside the MNO’s infrastructure results in a considerable decrease in RTT. Therefore, in order to allocate higher bandwidth to video contents with high bit-rate requirements, we cache high bit-rate contents closer to the end users, which increases their TCP throughput and consequently reduces latency. The quality priority factor estimates the summation of the bit-rates of cached contents normalized over sum of bit-rates of all video contents as follows:

\[
\Omega = \frac{\sum_{n=1}^{N} \sum_{i=1}^{I} \sum_{j=1}^{J} b_{ij} \cdot x_{nij}}{N \cdot \sum_{i=1}^{I} \sum_{j=1}^{J} b_{ij}} \tag{2}
\]

**Return Function** \( (\mathcal{R}) \) is the benefit gained from our virtualized caching system, which lays in the fact that caching video contents in the MNO’s infrastructure would minimize the traffic load that would be directed to public CDNs. As customers of these CDNs, content providers are charged on the basis of the amount of traffic that is served from the CDN. We assume that the benefit of transmission saving follows a predefined function \( \Gamma : \mathbb{R} \rightarrow \mathbb{R} \). Thus, we estimate the benefit derived from the reduction in transmission bandwidth (hereinafter offloaded traffic) when videos are cached for cluster \( n \) of the virtual caching system as

\[
\mathcal{R}(L_n) = \Gamma \left( \sum_{i=1}^{I} \sum_{j=1}^{J} l_{ij} \cdot x_{nij} \right) \quad \forall n \in \mathbb{N}. \tag{3}
\]

**Cost Function** \( (\mathcal{C}) \) is the cost incurred, which is represented by the amount of storage that is required for caching video contents. In general, public CDNs charge their customers based on the amount of bandwidth served by them. However, since the traffic load would traverse the MNO’s infrastructure whether or not the contents are cached, the main factor incurring cost would be the cost of storage. 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

\[
\mathcal{C}(S_n) = \Lambda \left( \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \cdot x_{nij} \right) \quad \forall n \in \mathbb{N}. \tag{4}
\]

Both benefit and cost functions can be any appropriate function defined by the operator. A summary of commonly used notation is provided in TABLE 1.

A salient assumption on the optimization problems defined in the sequel is that the consideration of a batch content processing. To this end, batch content pre-processing and the decision making of when to process and optimize popular content is implementation depended and as such can be deemed as beyond the scope of the paper.

### TABLE 1

<table>
<thead>
<tr>
<th>Commonly Used Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
</tr>
<tr>
<td>( i )</td>
</tr>
<tr>
<td>( j )</td>
</tr>
<tr>
<td>( n )</td>
</tr>
<tr>
<td>( I )</td>
</tr>
<tr>
<td>( J )</td>
</tr>
<tr>
<td>( N )</td>
</tr>
<tr>
<td>( q_{ij} )</td>
</tr>
<tr>
<td>( x_{nij} )</td>
</tr>
<tr>
<td>( f_{ij} )</td>
</tr>
<tr>
<td>( p_{ij} )</td>
</tr>
<tr>
<td>( b_{ij} )</td>
</tr>
<tr>
<td>( l_{ij} )</td>
</tr>
<tr>
<td>( \Omega )</td>
</tr>
<tr>
<td>( L_n )</td>
</tr>
<tr>
<td>( S_n )</td>
</tr>
<tr>
<td>( \mathcal{R} )</td>
</tr>
<tr>
<td>( \mathcal{C} )</td>
</tr>
<tr>
<td>( C_{\text{max}} )</td>
</tr>
<tr>
<td>( B_{\text{max}} )</td>
</tr>
</tbody>
</table>

### 3. PROBLEM FORMULATION

In this section, we formulate two virtual proactive caching problems based on the system model introduced in Section 2. The first optimization problem is formulated to achieve the optimal trade-off between the cost of caching video content (investment) and the benefit gained from content caching (return)\(^2\). The second optimization problem aims to maximize the offloaded traffic under the constraint of the total caching budget.

#### 3.1 Return on Investment Maximized Caching

We formulate the caching problem aimed at maximizing the return on investment [hereinafter referred to as maximum return on investment (MROI)] as follows:

\[
\max \frac{\sum_{n=1}^{N} \mathcal{R}(L_n)}{\sum_{n=1}^{N} \mathcal{C}(S_n)} \tag{5}
\]

subject to:

\[
\sum_{i=1}^{I} \sum_{j=1}^{J} b_{ij} \cdot x_{nij} \leq B_{\text{max}} \quad \forall n \in \mathbb{N} \tag{5a}
\]

\[
x_{nij} \geq x_{nij-1} \quad \forall n \in \mathbb{N}, \forall i \in J, \forall j \in J - \{1\} \tag{5b}
\]

\[
x_{nij} \in \{0, 1\} \quad \forall n \in \mathbb{N}, \forall i \in J, \forall j \in J. \tag{5c}
\]

The objective of optimization problem (5) is to find the optimal caching table \( x \) which determines what content should be cached for which cluster such that the ratio of overall return (3) to overall cost (4) is maximized. Constraint (5a) ensures that the sum of bit-rates of the video objects cached in the caching system for cluster \( n \) is upper-bounded by the maximum fronthaul capacity threshold, \( B_{\text{max}} \). This ensures adequate provision for the peak bandwidth. Constraint (5b) ensures that if a video quality layer is cached, all the lower quality layers are cached too (SVC requirement). We use binary variables \( x_{nij} \in \{0, 1\} \) explained in Section.

2. It is worth pointing out that the price of memory is decreasing by around 40% each year, whereas fronthaul/backhaul capacity leasing does not follow the same trend.
3.2 Budget-Constrained Caching

The budget-constrained caching problem, namely maximum offloaded traffic (MOT) is formulated as follows:

\[
\max_{x} \sum_{n=1}^{N} \sum_{i=1}^{I} \sum_{j=1}^{J} b_{ij} \cdot x_{nij} \tag{6}
\]

subject to:

\[
\sum_{n=1}^{N} c(S_n) \leq C^{\text{max}} \quad \forall n \in N \tag{6a}
\]

\[
\sum_{i=1}^{I} \sum_{j=1}^{J} b_{ij} \cdot p_{ij} \cdot x_{nij} \leq B_{n}^{\text{max}} \quad \forall n \in N \tag{6b}
\]

\[
x_{nij} - 1 \geq x_{nij} \quad \forall n \in N, \forall i \in I, \forall j \in J \tag{6c}
\]

\[
x_{nij} \in \{0, 1\} \quad \forall n \in N, \forall i \in I, \forall j \in J. \tag{6d}
\]

The objective of optimization problem (6) is to find the optimal allocation pattern \(x_n\) which maximizes the amount of cached traffic. (6a) represent the budget constraint. Constraints (6b)-(6d) are identical to the constraints in (5).

\[
Y^{n} = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1
\end{bmatrix} \tag{7}
\]

We define a cache indicator vector \(x \triangleq [x_n]_{N \times 1}\), where \(x_n = [x_{nij}]_{I \times 1}\). Each entry \(x_{nij} \in \{0, 1\}\) indicates whether the cache allocation pattern \(a\) is allocated for cluster \(n\) or not.

Note that all the clusters in the virtual caching system have the same cache allocation patterns matrix. We rewrite (5) as a BIP problem as follows:

\[
\min_{x} \{ p(x) = -\sum_{n=1}^{N} \sum_{a=1}^{A} c(S_n) \cdot x_{n} \} \tag{8}
\]

subject to:

\[
\sum_{a=1}^{A} b_{na} \cdot x_{na} \leq B_{n}^{\text{max}} \quad \forall n \in N \tag{8a}
\]

\[
x_{na} - \lfloor x_{na} \rfloor = 0 \quad \forall n \in N, \forall a \tag{8b}
\]

\[
\sum_{a=1}^{A} x_{na} \leq 1 \quad \forall n \in N. \tag{8c}
\]

where \(\Omega = \sum_{n=1}^{N} \sum_{a=1}^{A} (b_{na} \cdot x_{na}/B)\), \(b_{na}\) and \(c_{na}\) are the transmission bandwidth benefit and storage cost of allocating pattern \(a\) to cluster \(n\). For cluster \(n\), (8a) puts an upper-bound of \(B_{n}^{\text{max}}\) on the fronthaul bandwidth capacity, which is equivalent to (5a). Constraint (8b) is a pure binary constraint that ensures \(x_{na} \in \{0, 1\}\). (8c) ensures that at most one allocation pattern is chosen for each caching layer.

Although the optimization problem (8) is simpler and more tractable than (5), the solution is still exponentially complex.

4 Canonical Dual Framework

4.1 Dual Problem Formulation

We convert our BIP problem (8) into a continuous space canonical dual problem using CDT [26], [29], 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 [30]. However, due to additional constraints, our problem is more complex. A framework for solving source allocation BIP problems using CDT is given in [31], which will be extended to solve (8).

We define the feasible space for the primal problem (8) by \(Z_p = \{x \in \{0, 1\} \}^{A \times I} \). We temporarily relax the equality constraints (8b) and (8c) 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 \(\wedge (y) = (\delta, \beta, \tau, \sigma) \in \mathbb{R}^g\), which is a vector valued mapping where \(\mathbb{R}^g\) is the feasible space for \(y\), and

\[
\left\{ \begin{array}{ll}
\lambda = \sum_{a=1}^{A} b_{na} x_{na} - B_{n}^{\text{max}} N \times 1 \\
\mu = |x_{na} - (x_{na} - 1)| \times N \times 1 \\
\nu = \sum_{n=1}^{N} c(S_n) \times x_{n} - C_{\text{max}} N \times 1 \\
\end{array} \right. \tag{9}
\]

Therefore, the feasible space for \(y\) is defined by \(y_d = \mathbb{R}^{N} \times \mathbb{R} \times \mathbb{R}^{NA} \times \mathbb{R}^{N} \delta \leq 0, \mu \leq 0, \nu \leq 0. \mu = \sum_{n=1}^{N} \sum_{a=1}^{A} c(S_n) \cdot x_{na} - C_{\text{max}}\).

Next, we define the indicator function [30] as

\[
V(y) = \begin{bmatrix}
0 & \text{if } y \leq 0 \\
+\infty & \text{otherwise}
\end{bmatrix} \tag{10}
\]

We rewrite the primal problem (8) in the canonical form using indicator function (10) as follows:

\[
\min \{ V(\wedge(y) + p(x)) \}. \tag{11}
\]

We now define \(y^{\ast} = (\lambda^{\ast}, \mu^{\ast}, \nu^{\ast})\) as the vector of dual variables associated with the corresponding restrictions \(y \leq 0\). The feasible space for \(y^{\ast}\) is defined by \(y_d = \mathbb{R}^{N} \times \mathbb{R} \times \mathbb{R}^{NA} \times \mathbb{R}^{N} \delta \geq 0, \mu \geq 0, \nu \geq 0. \). Based on the Fenchel transformation, the canonical sup-conjugate function of \(V(y)\) is defined as

\[
V^{\ast}(y^{\ast}) = \sup \{ (y, y^{\ast}) - V(y) | y \in y_d, y^{\ast} \in y_d \} = \sup \{ (\lambda^{\ast} \lambda^{\ast} + \mu^{\ast} + \mu^{\ast} + \nu^{\ast} - V(y)) \}
\]

\[
\begin{cases}
0 & \text{if } \lambda^{\ast}, \mu^{\ast}, \nu^{\ast} \\
+\infty & \text{otherwise}
\end{cases} \tag{12}
\]

Using the definition of sub-differential, it can be easily verified that if \(y^{\ast} > 0\), then the condition \(y^{\ast} y = 0\) leads to \(y = 0\), and consequently \(x \in X_p\). Hence, the dual feasible space for the primal problem in (8) is an open positive cone defined by \(X^+_p = \{ y^{\ast} \in y_d | y^{\ast} > 0 \} \).

We define the total complementarity function [26] as

\[
\Xi(x, y^{\ast}) = \wedge(x)^{T} y^{\ast} - V^{\ast}(y^{\ast}) + p(x), \tag{13}
\]

which is obtained by replacing \(V(y) = \wedge(x)^{T} y^{\ast} - V^{\ast}(y^{\ast})\) (Fechnel-Young equality) in (11). We use the definitions of \(\wedge(x), V^{\ast}(y^{\ast})\) and \(p(x)\) to express \(\Xi(x, y^{\ast}) = \Xi(x, \lambda^{\ast}, \mu^{\ast}, \nu^{\ast})\) as given by (14), shown in the next page.
\[ \Xi(x, y^*) = \sum_{n} \sum_{a} \left[ x_{na} (\lambda_n b_{na} + \nu_n^* - \mu_n^*) \right] - \sum_{n=1}^{N} \lambda_n b_{n}^{\max} - \sum_{n=1}^{N} \nu_n^* + \sum_{n=1}^{N} \sum_{a} \mu_n^* x_{na}^2. \]  

Next, we define the canonical dual function [26], [30] using the canonical dual variables as

\[ \Upsilon(\lambda^*, \mu^*, \nu^*) = \text{sta} \{ \Xi(x, \lambda^*, \mu^*, \nu^*) \}, \]  

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

\[ x_{na}(y^*) = \frac{\partial + \zeta}{2\mu_n^*}, \quad \forall n, a, \]  

where \( \partial = [b_{na}, \rho(L_{na})] / [B \cdot \epsilon(S_{na})] \) and \( \zeta = \mu_n^* - \lambda_n^* b_{na} - \nu_n^* \). The stationary point is obtained through \( \nabla_x \Xi(x, y^*) = 0 \). Using (15) and (16), we obtain the dual function, which is given by (17), shown at the next page.

The dual function is a concave function on \( \mathcal{X}_p^* \). The canonical dual problem associated with (8) can be formulated as

\[ \min \{ \mathcal{P}(x) | x_p \} = \max \{ \Upsilon(\lambda^*, \mu^*, \nu^*) | x_p^* \}. \]  

**Theorem 1.** If \( \mathcal{P}(\tilde{x}) = \Upsilon(\tilde{y}^*) \) where \( \tilde{x} \) denotes the KKT point of the primal problem and \( \tilde{y}^* = (\tilde{\lambda}^*, \tilde{\mu}^*, \tilde{\nu}^*) \in \mathcal{X}_p^* \) denotes the KKT point of the dual function, there exists a perfect duality relationship between the primal problem in (8) and its canonical dual problem.

**Proof.** The proof directly extends from [29].

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

**Proof.** The proof directly extends from [29].

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 (17) is necessary and sufficient for global optimality as the dual problem is a concave maximization problem over \( \mathcal{X}_p^* \).

The KKT conditions of the dual function in (17) are given by \( \partial \Upsilon / \partial \lambda_n = 0 \), \( \partial \Upsilon / \partial \mu_n^* = 0 \), and \( \partial \Upsilon / \partial \nu_n^* = 0 \). (19), (20) and (21), shown in the next page, give the respective partial derivatives.

4.2 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 [27] algorithm for solving the complex non-linear equations associated with the KKT conditions [32]. 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 as follows:

- **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 \( \sigma_{\text{initial}} \) to a final value \( \sigma_{\text{final}} \) according to equation \( \sigma_{\text{iter}} = (\sigma_{\text{initial}} - \sigma_{\text{final}}) / \text{iter}_{\text{max}}^\text{iter} + \sigma_{\text{final}} \), where \( g \) is the modulation index;
- **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 [27], [33].

5 Simulation Results

We assume a cache-enabled cloud-based operator network consisting of four clusters. We evaluate the performance of our caching schemes in terms of return on investment, offloaded traffic, quality metric and cache size, which represent the gain achieved from the viewpoint of the content provider, MNO, end-user and MNO, respectively.

As in [34], [35], we assume that the video popularity is Zipf-like with a parameter of 0.65 and the video file sizes follow a Pareto (0.25) distribution with a minimum size of 60 megabytes. Without loss of generality, we suppose caching is performed at the level of entire video objects as in [36]. We can simply adjust index \( i \) to represent the \( i^{th} \) chunk rather than video object to enable caching at the chunk level.

We compare our proposed approach with the hit rate optimal caching algorithm LFU, which caches the most popular video contents [36], [37]. In contrast with the other widely used caching algorithm, LRU, LFU focuses on historical popularity over a long period of time. As a
\[
\Upsilon(X^*, \mu^*, \nu^*) = - \sum_{n=1}^{N} \sum_{a=1}^{A} \left[ \frac{\vartheta + \zeta}{2\mu_n^a} (3\vartheta + \zeta) \right] - \frac{N^A}{\sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_n^a \vartheta + \zeta}{2\mu_n^a} \right)} \cdot \frac{\sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_n^a \vartheta + \zeta}{2\mu_n^a} \right)}{B \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\zeta(S_{na} \vartheta + \zeta)}{2\mu_n^a} \right)} - \sum_{n=1}^{N} \lambda_n^a \mu_n^a \mu_n^a \mu_n^a - \nu^* \mu_n^a \mu_n^a \mu_n^a.
\]

(17)

\[
\frac{\partial \Upsilon}{\partial \lambda_n^a} = \sum_{n=1}^{N} \sum_{a=1}^{A} \left[ \frac{\mu_n^a \vartheta - \vartheta - \zeta}{4\mu_n^a} (3\vartheta + \zeta) + \frac{\mu_n^a \vartheta - \vartheta - \zeta}{4\mu_n^a} \right] - \frac{\sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_n^a \vartheta - \vartheta - \zeta}{2\mu_n^a} \right)}{B \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\zeta(S_{na} \vartheta + \zeta)}{2\mu_n^a} \right)} \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_n^a \vartheta - \vartheta - \zeta}{2\mu_n^a} \right) - \sum_{n=1}^{N} B_n^{max},
\]

(19)

\[
\frac{\partial \Upsilon}{\partial \mu_{na}^a} = \sum_{n=1}^{N} \sum_{a=1}^{A} \left[ \left( \frac{\mu_{na}^a - \vartheta - \zeta}{4\mu_{na}^a} \right) (3\vartheta + \zeta) - \frac{\mu_{na}^a - \vartheta - \zeta}{4\mu_{na}^a} \right] - \frac{\sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_{na}^a - \vartheta - \zeta}{2\mu_{na}^a} \right)}{B \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\zeta(S_{na} \vartheta + \zeta)}{2\mu_{na}^a} \right)} \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_{na}^a - \vartheta - \zeta}{2\mu_{na}^a} \right) - \sum_{n=1}^{N} B_n^{max},
\]

(20)

\[
\frac{\partial \Upsilon}{\partial \nu^*} = \sum_{n=1}^{N} \sum_{a=1}^{A} \left[ \frac{1}{4\mu_{na}^a} (3\vartheta + \zeta) + \frac{3\vartheta + 3\zeta}{4\mu_{na}^a} \right] - \frac{\sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_{na}^a - \vartheta - \zeta}{2\mu_{na}^a} \right)}{B \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\zeta(S_{na} \vartheta + \zeta)}{2\mu_{na}^a} \right)} \cdot \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{\mu_{na}^a - \vartheta - \zeta}{2\mu_{na}^a} \right) - \sum_{n=1}^{N} B_n^{max} - N.
\]

(21)

caching technique, our approach also considers a long term content popularity. Therefore, it is pertinent to compare our proposed scheme with LFU. Additionally, the results in [36] confirm the relative loss in hit rate of LRU compared with LFU observed for homogeneous content.

We consider three scenarios and measure the aforementioned metrics in each scenario. In Scenario 1, the total number of popular contents varies in the range [500,10000] (with 4 quality layers) whereas the sum fronthaul capacity is set to 25 Gbps. The default number of content is set as 4000 in scenarios 2 and 3. We vary the overall fronthaul capacity in the commercially available range of 15 to 40 Gbps (increments of 5 Gbps) in the former and the total cost from 0 to 1 in the latter. We relax the caching budget constraint in scenarios 1 and 2. A comparison of the performance of different caching techniques under the three scenarios is illustrated in TABLE 2.

In each of the above mentioned simulation scenarios, we solve the KKT conditions for each dual variable associated with the dual problem using the IWO algorithm (implemented in MATLAB) and compute the allocation vector \( x_n \) using (16). A pseudo code for the resource allocation algorithm is given as Algorithm 1. TABLE 3 provides a summary of the simulation parameters for IWO.

Fig. 2 shows the convergence of IWO algorithm for one of the KKT conditions \( \partial \Upsilon / \partial \lambda_n^a \). The x-axis shows the
Algorithm 1: C-RAN caching based on IWO (adapted from [32])

initialize $\lambda^*, \mu^*, \nu^*, \tau^*, \forall n \in N, \text{iter} = 0$;
$\forall \partial \mathcal{F}/\partial \nu^*$, where $\nu^* \in (\delta^*, \mu^*, \nu^*, \tau^*)$
create randomly dispersed initial population of $Q$ individuals (weeds):
$\mathcal{W} = \{W_1, \ldots, W_Q\}$;
while $|\nu^*| > \phi$ or $\text{iter} = \text{iter}_{\text{max}}$ do
    evaluate the fitness of each individual i.e.,
    calculate $f(W_n), \forall n \in \mathcal{W}$ and the colony’s best
    ($f_{\text{best}}$) and worst ($f_{\text{worst}}$) fitness;
    sort $\mathcal{W}$ in ascending order according to $f(W_n)$;
    select the first $Q_P$ individuals of $\mathcal{W}$ to create the set
    $\mathcal{W}_P$;
    Reproduction:
    $\forall W_j, j = 1, \ldots, Q_P$ generate
    $S_j = (f(W_j) - f_{\text{worst}} \times (S_{\text{max}} - S_{\text{min}}) + S_{\text{max}}$ seeds;
    create the population of the generated seeds,
    $\mathcal{W}_s = \{W_s\}$;
    Spatial Dispersion:
    for $i = 1 : |\mathcal{W}_s|$ do
        $W_s^i \leftarrow W_s^i + \phi^i$, where $\phi^i \sim L(0, \sigma_{\text{iter}})$;
    end
    Competitive Exclusion:
    create parents and seeds, $\mathcal{W}^* = \mathcal{W} \cup \mathcal{W}_s$;
    sort $\mathcal{W}^*$ in ascending order according to fitness;
    select the first $Q_{\text{max}}$ individuals of $\mathcal{W}^*$ and create
    $\mathcal{W}$;
end
select the best fitted individuals $\lambda^*, \mu^*, \nu^*$ and $\tau^*$;
calculate $x_n$ using (16);

TABLE 2
Performance Comparison of Caching Techniques

<table>
<thead>
<tr>
<th>Metric$^1$</th>
<th>Scenario</th>
<th>Figure</th>
<th>BPA$^2$</th>
<th>MRI (%)</th>
<th>MOT (%)</th>
<th>LFU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROI</td>
<td>1</td>
<td>Fig. 3(a)</td>
<td>MRI</td>
<td>+37.1</td>
<td>+33.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Fig. 4(a)</td>
<td>MRI</td>
<td>+38.2</td>
<td>+32.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Fig. 5(a)</td>
<td>MRI</td>
<td>+16.44</td>
<td>+30.87</td>
<td></td>
</tr>
<tr>
<td>CS</td>
<td>1</td>
<td>Fig. 3(b)</td>
<td>MRI</td>
<td>-42.27</td>
<td>-21.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Fig. 4(b)</td>
<td>MRI</td>
<td>-34.41</td>
<td>-17.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Fig. 5(b)</td>
<td>MRI</td>
<td>-16.22</td>
<td>-13.51</td>
<td></td>
</tr>
<tr>
<td>OT</td>
<td>1</td>
<td>Fig. 3(c)</td>
<td>MOT</td>
<td>+28.45</td>
<td>-34.72</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Fig. 4(c)</td>
<td>MOT</td>
<td>+25.7</td>
<td>-34.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Fig. 5(c)</td>
<td>MOT</td>
<td>+16.2</td>
<td>-28.9</td>
<td></td>
</tr>
<tr>
<td>QM</td>
<td>1</td>
<td>Fig. 3(d)</td>
<td>MRI</td>
<td>-13.01</td>
<td>+21.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Fig. 4(d)</td>
<td>MRI</td>
<td>+11.74</td>
<td>+21.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Fig. 5(d)</td>
<td>MRI</td>
<td>-12.67</td>
<td>+20.87</td>
<td></td>
</tr>
</tbody>
</table>

$^1$ROI: return on investment; CS: cache size; OT: offloaded traffic; QM: quality metric.
$^2$BPA: best performing algorithm.

Fig. 2. Convergence of IWO for the KKT condition $\partial \mathcal{F}/\partial \lambda_n^* = 0$.

number of iterations whereas the y-axis shows the value of fitness function, which is $\partial \mathcal{F}/\partial \lambda_n^*$. Over 500 iterations, the value of fitness function is $2.64 \times 10^{-06}$.

5.1 Scenario 1 - Variable Content Population

Fig. 3 demonstrates the performance of the caching algorithms under Scenarios 1. As shown in Fig. 3(a), for MRI, a growth in the size of the database initially decreases the return on investment. However, once the content population reaches a certain size ($\geq 6000$), it enters a steady state and remains unchanged. Before reaching a steady state, both overall cache size and offloaded traffic have an increasing behavior as the number of contents rises [see Fig. 3(b) and Fig. 3(c)]. However, the growth in the cache size incurs higher cost than the benefit gained from the increase in the offloaded traffic load, which leads to a gradual decrease in the return on investment. At the point that the return on investment reaches a steady state, the same occurs to cache size and offloaded traffic. This can be justified by the direct relationship between return on investment and the ratio of return function (related to offloaded traffic) and cost function (related to cache size).

As can be seen from Fig. 3(d), there is a slight positive correlation between return on investment and quality metric. As the content population increases, MRI demonstrates a tendency to cache more video objects in order to prevent...
the quality metric to be reduced significantly. This in turn decreases the return on investment due to the noticeable rise in cache storage cost.

Likewise, Fig. 3(a) indicates that in case of MOT and LFU, return on investment decreases alongside the increase in the size of the content database. However, due to the cost unawareness nature of the aforementioned schemes, they demonstrate a considerably higher decrease in return on investment in comparison with MRI. As shown in Fig. 3(c) and Fig. 3(b), by considering both the size and popularity of video contents and not taking storage into account, MOT induces the highest increase in offloaded traffic, and consequently cache storage requirements.

With SVC, in order to decode a higher video quality representation, all of the lower quality layers are needed. Therefore, low video quality layers which are smaller in size and bit-rate have greater popularity than high quality layers. Since LFU only takes popularity into consideration, it caches highly popular videos, which are normally smaller in size and bit-rate in comparison with higher quality representations. Hence, as shown in Fig. 3(b), it leads to lower storage requirements compared with MOT in addition to lower offloaded traffic [Fig. 3(c)] and quality [Fig. 3(d)].

5.2 Scenario 2 - Variable Fronthaul Capacity

Fig. 4 evaluates the performance of the caching schemes under Scenarios 2. Similar to Scenario 1, MRI outperforms both MOT and LFU with regard to return on investment, storage efficiency and quality. Likewise, the best performance in terms of increasing the offloaded traffic is achieved by MOT. As the fronthaul capacity increases, MRI also takes higher bit-rate video objects into account, which increases the quality metric significantly as shown in Fig. 4(d). Therefore, with the increase of the fronthaul capacity, at the cost of a slight reduction in the return on investment [Fig. 4(a)] and storage efficiency [Fig. 4(b)], we achieve a considerable increase in the quality metric [Fig. 4(d)] and a satisfactory rise in offloaded traffic load [Fig. 4(c)].

For MRI and LFU, increasing the fronthaul capacity relaxes the fronthaul capacity constraint, and hence enables caching more video contents. However, giving priority to video objects that are large in size and popularity, MOT leads to a higher increase in offloaded traffic compared with both MRI, which takes cost into consideration by maximizing the return on investment and LFU, which only considers popularity.

5.3 Scenario 3 - Variable Cost

Fig. 4 analyzes the performance of the caching schemes under Scenarios 3. Unlike the first and second scenarios where no caching budget constraint is set, here we consider maximum budget as the varying factor. Since MRI does not have a budget constraint [see (5)], varying the cache budget causes no change to its performance, and hence it demonstrates a static behavior. Similar to Scenarios 1 and 2, in this scenario, MRI has a better performance in terms of return on investment, storage efficiency and quality. Likewise, MOT results in a higher increase in offloaded traffic load.

For a fixed number of content items (4000), as the budget constraint increases, MOT continues to cache more contents. As shown in Fig. 4(c), giving priority to video objects which are large in size and popularity, MOT leads to a higher increase in offloaded traffic in comparison with MRI, which takes cost into consideration by maximizing the return on investment and LFU, which only considers popularity. This in turn causes MOT and MRI to have the lowest and highest storage efficiency, respectively [see Fig. 4(b)].

We note that after a certain increase in the budget, LFU reaches a steady state as it has already cached the contents with highest popularity. Caching more contents requires higher fronthaul capacity, which is set to 25 Gbps in this scenario. However, since MOT takes both popularity and size of the objects into consideration, further increase in the budget results in availability of more storage. This leads MOT to cache larger contents (consequently lower storage efficiency). However, it achieves a considerable gain in offloaded traffic. Having cached higher quality representations, MOT exhibits a better performance in terms of quality when compared to LFU.

5.4 Summary

TABLE 4 presents a comparison of the average performance of the three caching algorithms under all scenarios. In summary, MRI outperforms the other schemes in terms of return on investment, cache storage efficiency and quality. In comparison with MOT and LFU, MRI results in an average improvement of 30.58% and 32.23% in return on investment, 31.63% and 17.46% in storage efficiency and 12.47% and 21.43% in quality, respectively. On the other hand, MOT has the best performance with regard to the increase in overall offloaded traffic. It outperforms MRI by 23.45% and LFU by 32.7%.

5.5 Complexity and Optimality Analysis

Lastly, we discuss in this sub-section the complexity of the IWO algorithm and provide some further insights regarding the optimality or, in other words, the competitiveness of the solutions. IWO is an iterative algorithm and is used for each dual variable associated with the dual function in (17). In each iteration for \( \lambda^* \geq 0, \mu^* \geq 0, \nu^* \geq 0 \), we compute \( N, NA \) and \( N_v \) variables, respectively. Therefore, it has an overall worst case complexity of \( O(iter_{max} \cdot \{2N + NA\}) \) [32]. A comprehensive assessment of the performance of IWO algorithm in terms of convergence and computational time can be found in [27] and [33].
With respect to optimality, the use of such metaheuristic framework provides us, unavoidably, solutions that might not be optimal. However, as numerical investigations reveal the solutions found lead to significant additional improvements in the overall system performance compared to simple greedy based algorithms such as LFU. The high quality of the proposed solutions compared to current well-used greedy algorithms and the computational efficiency of finding them, strongly supports the potential application of the proposed framework in a real-world settings.

6 CONCLUSION

In this paper, we have proposed a CaaS framework for virtual caching in the MNO's infrastructure. Our first proposed scheme caches video contents in the cloud-based mobile network with the aim of maximizing the return on caching investment. Our second approach aims at maximizing the offloaded traffic as a result of caching for a given caching budget. We use CDT to convert our BIP virtual caching 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 framework outperforms LFU algorithm by more than 32%, 21%, 32% and 17% improvements in terms of return on investment, quality, offloaded traffic and storage efficiency, respectively.

A possible future avenue of research would be to consider the effect of cache sharing between different tenants in virtualized sliced network architectures where the cost as defined will change as well as the effect on fronthaul/backhaul sliced capacity per tenant. It is also interesting to understand the impact of temporal dynamics of access patterns and content popularity on the performance of the model and elaborate on the reactive mechanisms for on-the-fly adjustments of the caching strategies in accordance to dynamically changing conditions. Finally, the future iterations of the proposed model might feature transition towards distributed optimization where caching decisions are made in a decentralized fashion between a multitude of co-operating counterparts.
Fig. 4. Scenario 2 - varying fronthaul capacity: (a) return on investment; (b) cache size; (c) offloaded traffic; (d) quality metric.

REFERENCES


Fig. 5. Scenario 3 - varying caching budget: (a) return on investment; (b) cache size; (c) offloaded traffic; (d) quality metric.

[References]
[31] A. Ahmad and M. Assaad, “Polynomial-complexity optimal re-
source allocation framework for uplink SC-FDMA systems,” in
2011, pp. 1–5.
“Energy-efficient uplink resource allocation in LTE networks with
M2M/H2H co-existence under statistical QoS guarantees,” IEEE
[33] E. Pourjafari and H. Mojallali, “Solving nonlinear equations sys-
tems with a new approach based on invasive weed optimization
algorithm and clustering,” Swarm and Evolutionary Computation,
[34] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “Youtube traffic character-
ization: A view from the edge,” in ACM SIGCOMM Conf. Internet
alyzing the video popularity characteristics of large-scale user
generated content systems,” IEEE/ACM Trans. Netw., vol. 17, no. 5,
mix on caching performance in a content-centric network,” in
October 1999.
We greatly appreciate the Editors’ attentive evaluation and recommendation for revision and the valuable comments from the anonymous reviewers. Based on the suggestions, we have carefully revised the paper and provided point-to-point responses to the reviewers’ comments to this end. Please refer to the following detailed responses in the sequel. The comments from the reviewers (including those from the Associate Editor) are marked with blue color for ease of reference.

ASSOCIATE EDITOR

Comments to the Authors

This paper studies CaaS and proposes some heuristics for solving two different version of this problem. The contributions in current format is limited as the current version mainly formulate the problem and then use some existing numerical based approaches to find a solution. The technology is somewhat standard and the analysis of the solution (complexity and optimality) is lacking.
Please address these comments, as explained in the reviews below. Since this is a second Major Revision, failure to address these comments in a satisfactory manner will result in rejecting the paper.

Response: Let us reiterate that we sincerely thank the Editor for the time taken to review all comments/suggestions and provide a synopsis regarding this second round of reviews for the manuscript. We have now strived to highlight the novelty of the research work and its applicability. Please note that we have included a separate subsection on complexity and optimality analyses (Section 5.5) of the proposed solution methodology (previously, the aforementioned section focused solely on complexity analysis). Furthermore, in order to provide a more holistic treatment, we also direct the interested reader to further literature on that very specific area.

The salient contribution of the work we believe is that the solution methodology (canonical dual programming and meta heuristic optimization) will inspire other researchers to use this framework to attack inherently complex nonlinear optimization problems that might be encountered in network optimization (for example in the areas of network slicing and VNF location, scaling and sharing, to mention just a few). We have provided a more detailed explanation in our responses to Reviewer 3’s comments.

Finally, we would also like to bring the Editor’s attention to a number of inconsistencies with Reviewer 3’s current comments compared to what have been mentioned by the same reviewer in the previous round of reviews. We highlight those in the relevant sections and encourage the Editor to pay special attention since those have - to a degree - hindered our revision.

**REVIEWER 1**

*Recommendation: Accept with no changes*

My comments have been addressed satisfactorily.

Response: We appreciate the reviewer’s positive view on our research work. We had strived in the previous review to take on board all comments. Once again, we thank the reviewer for providing detailed and valuable comments for us to improve the paper.

**REVIEWER 2**

*Recommendation: Accept with no changes*

The concerns I raised in the previous version were addressed. The current manuscript shows significant improvement and I have no more problems.

Response: We appreciate the reviewer’s positive view on our research work. We had strived in the previous review to take on board all comments. Once again, we thank the reviewer for providing detailed and valuable comments for us to improve the paper.
**Reviewer 3**

**Recommendation:** Reject

**Comments**

1) The paper is extremely hard to follow. There are a lot of abbreviations, most of them are totally unnecessary. Moreover, numerous concepts are proposed without any definition, like “quality level”, “operator networks”, “quality of experience”, “fronthaul capacity” and so on. The mathematical part is also extremely hard to follow, as the authors almost did not provide any intuitive explanations on the proposed solution. Due to the extremely casual and unintelligent style of the paper’s presentation, I think that the paper is neither interesting nor convincing.

Response: We thank the reviewer for the above comment. To take on board the above suggestions in this revised version we have removed some of the acronyms which are very well-known in the community such as the base station (BS), service provider (SP), content provider (CP) and core network (CN). We have also formally added a description to some well-known concepts such as quality of experience (QoE) where we have added as a footnote a document from ITU which contains the formal definition of this concept. Based on those changes we believe now the paper contains adequate number of acronyms taking into account the scope area of research and length of the manuscript. Please also note below the comment from the previous round of reviews that we have already addressed.

Comment from Reviewer 3 in the first round of reviews: ”There are a lot of abbreviations in this paper, the paper needs to add a table to summarize it.”

2) The problem model seems to be very artificial. The paper imagines a scenario characterized by numerous known parameters (as those listed in Table 1), and there are no justifications on whether the system model and the parameters are realistic. Moreover, only some simulations are provided to evaluate the effectiveness of the proposed approach. Therefore, I do not believe that the problem studied in the paper actually exists in a real mobile cloud system, unless the authors can build a testbed to prove that.

Response: Thanks to the reviewer for this comment. Very similar system model and system parameters have been also used the literature, such as Li et al. [20] and Zhang et al. [21] cited below. We also assume a set of popular content with the aim to finding optimal trade-offs between caching and cost of the cache (since caching all contents is infeasible). The main difference being that they assume a fully distributed caching environment at the level of the base stations whereas in our paper we assume a virtualized edge cloud based approach.

3) The novelty of the techniques adopted in the paper is also not clear. It seems that some existing optimization tools (such as the invasive weed optimization method) are directly applied to solve the problem formulated in the paper.

Response: As we have highlighted in the Introductory section of the paper, there are multifaceted contributions in the paper. In that promising area of mobile edge caching, which is still nascent, we provided a novel mathematical programming formulation that takes into account the return on investment as well as budget constrained caching. We would like to stress all along the line that these two types of mathematical programming formulation have not been considered previously in the literature and as such provide interesting insights into the scope area of research. The novelty of the problem formulation has been stressed in the paper and elude the differences with the state of art in the scope area of research. Some commenting hereafter regarding the solution methodology. We have indeed used recent advanced metaheuristic algorithm to attack our proposed optimization problem. A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. In that respect, the algorithm need to be adjusted to the specific problem at hand. We have indeed bring the problem to a suitable stage after the canonical dual problem and the KKT conditions so that to apply the Invasive Weed Optimization framework. Metaheuristic frameworks (such as genetic algorithms, particle swarm optimization etc.) have been extensively used within the IEEE community even though their roots come from the applied maths and operations research communities; however, as we have already mentioned they are powerful framework for finding competitive solutions in hard optimizations problems and require special attention to be used in specific general optimization problems. Metaheuristic frameworks are defined in general terms, and can be adapted to fit the needs real-world optimization problems. However, this flexibility comes at the cost of requiring considerable problem-specific adaptation to achieve good performance; and this is part of the calibration that need to be done. We hope the way that we have used dual canonical optimization framework in conjunction with the advanced metaheuristic optimization technique might inspire other researchers to use such powerful techniques in their specific field of network optimization and mobile computing.

**Additional Questions**

1. How relevant is this manuscript to the readers of this periodical? Please explain under Public Comments below.: Interesting - but not very relevant
Response: Thanks to the reviewer for the above comment. We would like to point out that in the previous round of reviews a different answered has been given, as shown below.

Comment from Reviewer 3 in the first round of reviews: “Relevant”

2. Is the manuscript technically sound? Please explain under Public Comments below.: No

Response: Thanks to the reviewer for the above comment. We would like to point out that we took into account and addressed the reviewer’s following comment in the previous round of reviews.

Comment from Reviewer 3 in the first round of reviews: “Partially”

3. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: Important references are missing; more references are needed

Response: Thanks to the reviewer for the above comment. To take into account this comment, we have added Zhang et al. [21] (cited below) to the manuscript. We would also like to point out that we had removed a number of references as per reviewer’s comment in the previous round of reviews, shown below.


Comment from Reviewer 3 in the first round of reviews: “Number of references are excessive”

4. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain under Public Comments below.: Could be improved

Response: Thanks to the reviewer for the above comment. We would like to point out that in the previous round of reviews a different answered has been given, as shown below.

Comment from Reviewer 3 in the first round of reviews: “Yes”

5. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Poor

Response: Thanks to the reviewer for the above comment. We would like to point out that in the previous round of reviews a different answered has been given, as shown below.

Comment from Reviewer 3 in the first round of reviews: “Satisfactory”

6. Please rate the readability of this manuscript. Please explain your rating under Public Comments below.: Unreadable

Response: Thanks to the reviewer for the above comment. We would like to point out that in the previous round of reviews a different answered has been given, as shown below.
Comment from Reviewer 3 in the first round of reviews: "Easy to read"

7. If this manuscript is an extended version of a conference publication, does it offer substantive novel contributions beyond those of the previously published work(s) i.e. expansion of key ideas, examples, elaborations etc. *New results are not required*: Not applicable
   Comment from Reviewer 3 in the first round of reviews: "Yes"

8. Please rate the manuscript. Please explain under Public Comments below.: Poor
   Response: Thanks to the reviewer for the above comment. We would like to point out that in the previous round of reviews a different answer has been given, as shown below.
   Comment from Reviewer 3 in the first round of reviews: "Good"
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 $\mathcal{I} \triangleq \{1, \ldots, I\}$. We index different quality layers of a video stream by the set $\mathcal{J} \triangleq \{1, \ldots, J\}$. By $q_{ij}$, we denote the $j^{th}$ quality layer of video $i$, which
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 \( N \triangleq \{1, \ldots, n, \ldots, N\} \). 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

\[
b_{n}(l_{nij}) = \Gamma \left( \sum_{i=1}^{I} \sum_{j=1}^{J} l_{nij} \right) \quad \forall n \in \mathbb{N}. \tag{1}
\]

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

\[
c_{n}(s_{n}) = \Lambda(s_{n}) \quad \forall n \in \mathbb{N}. \tag{2}
\]

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.

\[
\max \times \sum_{n=1}^{N} b_{n}(l_{nij}) \\
\sum_{n=1}^{N} c_{n}(s_{n}) \tag{3}
\]

subject to:

\[
s_{n} = \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} x_{nij} \leq m_{n} \quad \forall n \in \mathbb{N} \tag{3a}
\]

\[
\sum_{n=1}^{N} x_{nij} \leq 1 \quad \forall i \in J, \forall j \in J \tag{3b}
\]

\[
x_{nij} - x_{nij-1} \geq 0 \quad \forall n \in \mathbb{N}, \forall i \in J, \forall j \in J - \{1\} \tag{3c}
\]

\[
x_{nij} \in \{0,1\} \quad \forall i \in J, \forall j \in J, \forall n \in \mathbb{N}. \tag{3d}
\]

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 \( \mathcal{K} \triangleq \{1, \ldots, K\} \). 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.

We define a cache indicator vector \( x = [x_{n}]_{N \times 1} \), where \( x_{n} = [x_{nij}]_{A \times 1} \). 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:

$$\min_x \left\{ \mathcal{P}(x) = \frac{1}{2} \sum_{n=1}^{N} \sum_{a=1}^{A} b_{na} x_{na} \right\}$$

subject to:

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

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

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

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

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 $Y^{n}$, 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 $X_{p} = \{ x \in \{0, 1\}^{NA} \}$. 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 $\wedge(y) = (\delta, \beta, \tau, \sigma) \in \mathcal{Y}_{g}$, which is a vector valued mapping where $\mathcal{Y}_{g}$ is the feasible space for $y$, and

$$\begin{align*}
\lambda &= \left[ \sum_{a=1}^{A} s_{na} x_{na} - m_{n} \right]_{N \times 1} \\
\mu &= \left[ \sum_{n=1}^{N} \sum_{a=1}^{A} Y_{ka}^{n} x_{na} - 1 \right]_{K \times 1} \\
\nu &= [x_{na} (x_{na} - 1)]_{N \times 1} \\
\tau &= [x_{na} (x_{na} - 1)]_{N \times 1}
\end{align*}$$

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

Next, we define the indicator function [15] as

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

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

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

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

$$V^{\ast}(y^{\ast}) = \sup \{ \langle y, y^{\ast} \rangle - V(y) | y \in \mathcal{Y}_{g}, y^{\ast} \in \mathcal{Y}_{d} \}$$

$$= \sup \{ \langle \lambda^{T} \lambda^{\ast} + \mu^{T} \mu^{\ast} + \nu^{T} \nu^{\ast} + \tau^{T} \tau^{\ast} - y_{g} \rangle \}$$

$$= \begin{cases} 
0 & \text{if } \lambda^{\ast}, \mu^{\ast}, \nu^{\ast}, \tau^{\ast} \geq 0 \\
+\infty & \text{otherwise.}
\end{cases}$$

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

We define the total complementarity function [11] as

$$\Xi(x, y^{\ast}) = \wedge(y^{\ast})^{T} y^{\ast} - V^{\ast}(y^{\ast}) + \mathcal{P}(x),$$

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

$$\Xi(x, y^{\ast}) = \sum_{n=1}^{N} \sum_{a=1}^{A} x_{na} \Phi - \sum_{n=1}^{N} \sum_{a=1}^{A} b_{na} x_{na}$$

$$- \sum_{n=1}^{N} \lambda_{n}^{T} m_{n} - \sum_{k=1}^{K} \mu_{k}^{T} \tau_{k}^{\ast} + \sum_{n=1}^{N} \sum_{a=1}^{A} \nu_{na}^{T} \nu_{na}^{2},$$

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

$$\Upsilon(\lambda^{\ast}, \mu^{\ast}, \nu^{\ast}, \tau^{\ast}) = \text{sta} \{ \Xi(x, \lambda^{\ast}, \mu^{\ast}, \nu^{\ast}, \tau^{\ast}) \},$$

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

\[
x_{na}(y^*) = \frac{1}{2} - \frac{1}{2
u_{na}} \left( \sum_{k=1}^{K} \mu_k Y_{kn} + \lambda_n s_{na} + \tau_n \right)
\]

\( \forall n, a \),

(13)

where the stationary point is obtained through \( \nabla_x \Xi(x, y^*) = 0 \). 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 \( X^*_p \). The canonical dual problem associated with (5) can be formulated as

\[
\min \{ \mathcal{P}(x) | x_p \} = \max \{ \Upsilon(\lambda^*, \mu^*, \nu^*, \tau^*) | x^*_p \}. 
\]

(15)

**Theorem 1.** If \( \mathcal{P}(\tilde{x}) = \Upsilon(\tilde{y}) \) where \( \tilde{x} \) denotes the KKT point of the primal problem and \( \tilde{y}^* = (\tilde{\lambda}^*, \tilde{\mu}^*, \tilde{\nu}^*, \tilde{\tau}^*) \in X^*_p \) 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 \( \tilde{y}^* = (\tilde{\lambda}^*, \tilde{\mu}^*, \tilde{\nu}^*, \tilde{\tau}^*) \in X^*_p \), then \( \tilde{x} \) is a global minimizer of \( \mathcal{P}(x) \) over \( X_p \) and \( \tilde{y}^* \) is a global maximizer of \( \Upsilon(\lambda^*, \mu^*, \nu^*, \tau^*) \) over \( X^*_p \). Hence, \( \mathcal{P}(\tilde{x}) = \min \{ \mathcal{P}(x) | x_p \} = \max \{ \Upsilon(\lambda^*, \mu^*, \nu^*, \tau^*) | x^*_p \} = \Upsilon(\tilde{\lambda}^*, \tilde{\mu}^*, \tilde{\nu}^*, \tilde{\tau}^*) \).

**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 \( X^*_p \).

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 \( \sigma_{initial} \) to a final value \( \sigma_{final} \) according to equation \( \sigma_{iter} = [(iter_{max} - iter)/iter_{max}]^g \sigma_{initial} + \sigma_{final} \), 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 \( \lambda^*, \mu^*, \nu^*, \tau^*, \forall n \in N, iter = 0; \)

\( \forall \mathcal{P}(\tilde{x}) > 0 \) or \( iter = iter_{max} \) do

evaluate the fitness of each individual i.e., calculate \( f(W_n), \forall n \in \mathcal{W} \);

sort \( \mathcal{W} \) in ascending order according to \( f(W_n) \);

select the first \( Q_p \) individuals of \( \mathcal{W} \) to create the set \( \mathcal{W}_p \);

\( \forall W_j, j = 1, \cdots, Q_p \) generate

\( S_j = f(W_j) - f_{\text{max}} \times (S_{\text{max}} - S_{\text{min}}) + S_{\text{max}} \) seeds;

create population of generated seeds, \( \mathcal{W}_s = \{ W_s \} \);

for \( i = 1: |\mathcal{W}_s| \) do

\( W_i^* \leftarrow W_i^* + \phi' \), where \( \phi' \sim L(0, \sigma_{\text{iter}}) \);

end

create \( \mathcal{W}^* = \mathcal{W} \cup \mathcal{W}_s \);

sort \( \mathcal{W}^* \) in ascending order according to fitness;

select the first \( Q_{\text{max}} \) individuals of \( \mathcal{W}^* \) and create \( \mathcal{W} \);

end

select the best fitted individuals \( \lambda^*, \mu^*, \nu^* \) and \( \tau^* \);

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 fourth 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
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 $x_k$ 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.

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

\[
\begin{align*}
Y(A, \mu, \nu, \tau) &= - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{\Phi^2}{2\nu_{na}^2} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{\nu_{na} \Phi}{2\nu_{na}} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{\nu_{na} \Phi}{2\nu_{na}} - \frac{N}{N} \sum_{n=1}^{N} \lambda_{n} m_{n} - \frac{N}{N} \mu_{k} - \frac{N}{N} \tau_{n}.
\end{align*}
\]

\[
\begin{align*}
\frac{\partial Y}{\partial \lambda_{n}} &= - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{s_{na} \Phi}{2\nu_{na}^2} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{b_{na} \Phi}{2\nu_{na}} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{c_{na} \Phi}{2\nu_{na}} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{b_{na} \Phi}{2\nu_{na}} - \sum_{n=1}^{N} \mu_{k} - \sum_{n=1}^{N} \nu_{na}.
\end{align*}
\]

\[
\begin{align*}
\frac{\partial Y}{\partial \nu_{na}} &= - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \left( \frac{K}{2\nu_{na}^2} \right) - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{b_{na} \Phi}{2\nu_{na}} + \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{c_{na} \Phi}{2\nu_{na}} - \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{b_{na} \Phi}{2\nu_{na}} - \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{c_{na} \Phi}{2\nu_{na}} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{c_{na} \Phi}{2\nu_{na}} - K,
\end{align*}
\]

\[
\begin{align*}
\frac{\partial Y}{\partial \tau_{n}} &= - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{\Phi}{2\nu_{na}^2} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{b_{na} \Phi}{2\nu_{na}} + \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{c_{na} \Phi}{2\nu_{na}} - \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{b_{na} \Phi}{2\nu_{na}} + \frac{N}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} \frac{c_{na} \Phi}{2\nu_{na}} - N.
\end{align*}
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of initial population ($Q$)</td>
<td>20</td>
</tr>
<tr>
<td>Min. fitness threshold ($\rho$)</td>
<td>$10^{-7}$</td>
</tr>
<tr>
<td>Max. no. of iterations ($\text{iter}_{\text{max}}$)</td>
<td>500</td>
</tr>
<tr>
<td>Max. no. of plants ($Q_{\text{max}}$)</td>
<td>10</td>
</tr>
<tr>
<td>No. of seeds ($S_{\text{max}}, S_{\text{min}}$)</td>
<td>(5,0)</td>
</tr>
<tr>
<td>Non-linear modulation index</td>
<td>2.5</td>
</tr>
<tr>
<td>Standard deviation ($\sigma_{\text{initial}}, \sigma_{\text{final}}$)</td>
<td>(10,0.01)</td>
</tr>
</tbody>
</table>
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: Provisioned 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 $\lambda^* \geq 0$, $\mu^* \geq 0$, $\nu^* \geq 0$, $\tau^* \geq 0$, we compute $N, K, NA$, and $N$ variables, respectively. Therefore, it has an overall worst case complexity of $O(\text{iter}_{\text{max}} \cdot (2N + K + NA))$. 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