2016

An Optimal Content Caching Framework for Utility Maximization

Ran Bi
the Department of Computer Science, Dalian University of Technology, Dalian 116024, China.

Yingshu Li
the Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA.

Xu Zheng
the Department of Computer Science, Harbin Institute of Technology, Harbin 150000, China.

Follow this and additional works at: https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology

Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation
Ran Bi, Yingshu Li, Xu Zheng. An Optimal Content Caching Framework for Utility Maximization. Tsinghua Science and Technology 2016, 21(4): 374-384.

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in Tsinghua Science and Technology by an authorized editor of Tsinghua University Press: Journals Publishing.
An Optimal Content Caching Framework for Utility Maximization

Ran Bi, Yingshu Li, and Xu Zheng

Abstract: For desirable quality of service, content providers aim at covering content requests by large network caches. Content caching has been considered as a fundamental module in network architecture. There exist few studies on the optimization of content caching. Most existing works focus on the design of content measurement, and the cached content is replaced by a new one based on the given metric. Therefore, the performance for service provision with multiple levels is decreased. This paper investigates the problem of finding optimal timer for each content. According to the given timer, the caching policies determine whether to cache a content and which existing content should be replaced, when a content miss occurs. Aiming to maximize the aggregate utility with capacity constraint, this problem is formalized as an integer optimization problem. A linear programming based approximation algorithm is proposed, and the approximation ratio is proved. Furthermore, the problem of content caching with relaxed constraints is given. A Lagrange multiplier based approximation algorithm with polynomial time complexity is proposed. Experimental results show that the proposed algorithms have better performance.

Key words: content caching; utility maximization; integer optimization; approximation algorithm

1 Introduction

The rapid increase in data consumption has been witnessed recently, due to the prevailing multimedia services[1, 2]. Larger amount of mobile users are continuously entertained in a wide variety of multimedia applications by smart phones and tablets[3, 4]. High-quality video is also demanded with the prevalence of high definition video. For desirable Quality of Service (QoS), content providers aim at covering millions of content requests in time by large network caches. In general, when a data item is first requested, it is temporarily stored at some nodes and the following requests are responded with these local copies. This paradigm serves users by the content cached in the local storage node, which reduces the overhead of limited network resources, such as bandwidth and server load. Hence a better quality of service can be provided.

Content caching has been considered as a fundamental module[5]. For example, cloud providers conduct caching services and many caching systems have been set up to accommodate vast data traffic[6]. QoS provisioning by the underlying networks aims to achieve end users’ satisfaction with low cost of network resource. There exist a variety of applications available for users, then different QoS is highly desirable, which depends on the content publisher and consumer.

Many efforts have been made to improve the data delivery, such as routing strategy[7], bandwidth sharing[8], and server placement[9]. However, there exist few studies on the optimization of content caching. The main methodology applied in the existing policies for cache management is to replace the cached content by a new one based on the given metrics to improve hit rate, such as Least Recently Used (LRU) and Least Frequently Used (LFU). These algorithms treat
differentiated contents by strongly coupling, so that the performance for service provision with multiple levels is decreased. Recently utility-based caching framework is proposed in Ref. [10]. However, this work is restricted in Time-To-Live (TTL) caches.

In this paper, a utility-driven caching strategy is proposed, where each content is assigned a timer. According to the given timer, the caching policies determine whether to cache a content and which existing content should be replaced, when a content miss occurs. The problem of finding optimal timer for each content is formalized as an integer optimization problem first, which aims at maximizing the aggregate utility with capacity constraint. In view of the hardness of the proposed problem, a linear programming based approximation algorithm is given and the approximation ratio is proved. There exists high diversity among popular contents, thus one content can not be kept all the time in view of utility gain. Furthermore, we propose the problem of content caching with relaxed constraints, which loosens the range of content timer. And we propose a Lagrange multiplier based approximation algorithm with polynomial time complexity. The main contributions of the paper are as follows.

- The problem of assigning optimal timer for each content is formulated as an integer optimization problem, where the objective is to maximize the aggregate utility with capacity constraint. The hardness of the problem is analyzed.
- A linear programming problem is proposed, which is equivalent to the problem of content timer assignment in the case of real numbers. Then a linear programming based approximation algorithm is given. The upper bound of approximation ratio is proved.
- The problem of content caching with relaxed constraints is given. And a Lagrange multiplier based approximation algorithm is provided, and the time complexity is $O \left( K T n \sum_{i \in \{1,2,\ldots,n\}} \{ p_i \lambda_i \} \right)$.
- Simulation experiments are performed to demonstrate the effectiveness of the proposed algorithms. Experimental results show that our algorithms achieve better performance in terms of utility gain and hit rate.

The rest of this paper is organized as follows. The related work on content caching is surveyed in Section 2. In Section 3, the problem is described. Section 4 gives a linear programming based approximation algorithm and the ratio bound is proved. A Lagrange multiplier based algorithm for content timer with relaxed constraints is provided in Section 5. Experimental results are illustrated in Section 6, and Section 7 concludes this paper.

2 Related Work

The content available at nearby caches is served locally, when a content is requested by end users. Content caching reduces the cost of data packet delivery in the network. Hence, content caching has important effect on the network performance, such as Information Centric Networking (ICN) [11], Content-Centric Network (CCN) [12], and Content Distribution Network (CDN). LRU and LFU are the most applied strategies in current network architecture, since they are easy to carry out. In LRU, the least recently requested content is replaced by the request content. And the content with least request number is replaced by the new one in LFU. However, both of the schemes overlook the tradeoff between hit rate and fairness, which are hard to achieve QoS provision with multi-level.

Reference [5] provides an online popularity learning algorithm, and presents a popularity-based content replacement strategy. To balance performance and caching cost, Ref. [13] proposed an online content migration strategy and a request distribution algorithm. In Ref. [14], the optimal content replacement without the prior knowledge of popularity is formalized as a multi-armed bandit problem. Famaey et al. [15] predicted the request pattern based on the predetermined models, and then a predictive popularity based content replacement algorithm is proposed. Reference [16] utilizes content routers to distributed measure a content popularity for CCN, and then proposes a popularity-based collaborative forwarding and caching scheme.

The growth of mobile computing imposes increasing demand on content caching [17, 18]. Maddah-Ali and Niesen [19] proposed a coded caching scheme for a single-hop broadcast network, which aims at requesting multiple users by coded-multicasting. To balance the cost among base station transmission, access point storage, and user connection, Ref. [20] provides a coded based caching strategy for heterogeneous wireless networks. To reduce the service delays, Ref. [21]
groups users into different clusters based on the social similarities, and then the content is elaborately requested by the small cell base stations. To reduce the accessing latency, Ref. [22] proposes threshold-based caching policy for wireless access networks, which caches all contents requested more times than the given threshold. Dehghan et al. [23] considered a joint caching and routing problem, which determines the optimal routes as well as the optimal caching policy.

Most of the existing work focus on the measurement of content popularity to improve the network performance. These algorithms treat differentiated contents by strongly coupling, that decreases the efficiency for multi-level QoS provision. Utility functions are widely used to optimize the network performance, such as congestion control [24], scheduling [25], throughput maximization [26], etc. Recently utility-based caching framework is proposed in Ref. [10]. However, this work is restricted in TTL caches.

In this paper, a utility-driven caching strategy is proposed, where each content is assigned a timer. According to the given timer, the caching policies determine whether to cache a content and which existing content should be replaced, when a content miss occurs.

3 Problem Description

In caching system, the caching policies determine whether to cache a content and which existing content should be replaced, when a content miss occurs. In this section, the problem of utility-driven content caching is formalized as a nonlinear integer optimization problem, and the hardness of the problem is NP-complete at least.

3.1 Problem definition

It is supposed that the content provider holds a set of contents, denoted by $C = \{c_1, c_2, ..., c_n\}$. And any content $c_i$ in set $C$ can be requested by end users. In practical applications, the set may contain millions of contents. For desirable quality of service, the caching system is set up, which utilizes large network caches to cover millions of populated content requests for users. In the caching system, each cache node implements independently, then we focus on the content caching scheme on a single local node. We assume that each content has the same size $3$, and each one takes up one unit of capacity. If the storage capacity of the cache node is $B$, the node can hold up no more than $B$ contents at time $t$. For given time interval $[T_1, T_1 + T]$ and content $c_i$, the process of requesting $c_i$ follows Poisson Process over $[T_1, T_1 + T]$ [5]. Thus the probability of content $c_i$ being requested $k$ times satisfies the following equation.

$$\Pr\{N(T_1 + T) - N(T_1) = k\} = \frac{e^{-\lambda T} (\lambda T)^k}{k!} \tag{1}$$

In a TTL cache, each content $c_i$ is associated with timer $t_i$. When the miss of content $c_i$ occurs, $c_i$ is stored in the cache and $t_i$ decreases with time slot. Content $c_i$ is removed when $t_i$ reduces to zero. Inspired by TTL caches, for $\forall i \in \{1, 2, ..., n\}$, $c_i$ can be represented by a tuple $\{p_i, \lambda_i, t_i\}$, where $p_i$ is the popularity weight of the content, $\lambda_i$ is the parameter of poisson process, and $t_i$ is the timer of content $c_i$ in the given time interval $[T_1, T_1 + T]$.

If the timer of content $c_i$ is set to $t_i$, according to Eq. (1) the probability of $c_i$ being requested over $[t, t + t_i]$ is as follows.

$$\Pr\{N(t + t_i) - N(t) \geq 1\} = 1 - e^{-\lambda_i t_i} \tag{2}$$

To quantify the caching efficiency and fairness, we consider the family of $\beta$-fair utility function [27]. By caching content $c_i$ over $t_i$ time slots, the utility is obtained as Formula (3), in which $\beta \in (0, 1)$.

$$u_i(t_i) = p_i \frac{(1 - e^{-\lambda_i t_i})^{1-\beta}}{1 - \beta} \tag{3}$$

In case of $\beta \in (1, +\infty)$, the utility function is defined as follows.

$$u_i(t_i) = -p_i \frac{(1 - e^{-\lambda_i t_i})^{1-\beta}}{1 - \beta} \tag{4}$$

Based on the analysis presented in this paper, the proposed algorithms are also applied to the case of $\beta \in (1, +\infty)$. Thus we only focus on the case of $\beta \in (0, 1)$ in the following.

We aggregate the utility of all contents, and aim at maximizing the summation of the utility with capacity constraint in given time interval $[T_1, T_1 + T]$. Based on the above analysis, the problem of finding optimal timer for each content can be formulated as the following integer optimization problem.

$$\max \sum_{i=1}^{n} p_i \frac{(1 - e^{-\lambda_i t_i})^{1-\beta}}{1 - \beta}$$

s.t. $\sum_{i=1}^{n} t_i \leq BT$, $0 \leq t_i \leq T$, $t_i \in Z, \forall i \in \{1, 2, ..., n\}$

The caching node with capacity $B$ can buffer $B$ contents at time $t$, thus the summation of timers is no
more than $BT$. The second constraint implies that the timer of each content is less than $T$.

### 3.2 Time complexity analysis

In the following, we show that the hardness of integer optimization problem (5) is no less than NP-complete.

**Lemma 1** For all $i \in \{1, 2, ..., n\}$, utility function $u_i(\cdot)$ is non-decreasing and concave, in which $p_i > 0$, $\beta \in (0, 1)$.

**Proof** According to the definition of utility function, it is easily known that for all $i \in \{1, 2, ..., n\}$, $u_i(\cdot)$ is non-negative in the case of $\tau_i \in (0, \infty)$. The first derivative of $u_i(\cdot)$ with respect to $\tau_i$ can be derived as follows.

$$\frac{\partial u_i}{\partial \tau_i} = p_i \lambda_i e^{-\lambda_i \tau_i} (1 - e^{-\lambda_i \tau_i})^{-\beta}$$

(6)

Based on the above formula, it can be known that the following is true.

$$\frac{\partial u_i}{\partial \tau_i} > 0.$$  

Thus utility function $u_i(\cdot)$ is non-negative and increasing. Based on Eq. (6), the second derivative of $u_i(\cdot)$ with respect to $\tau_i$ satisfies the following.

$$\frac{\partial^2 u_i}{\partial \tau_i^2} = -p_i \lambda_i^2 e^{-\lambda_i \tau_i} (1 - e^{-\lambda_i \tau_i})^{-\beta} - \beta p_i \lambda_i^2 e^{-2\lambda_i \tau_i} (1 - e^{-\lambda_i \tau_i})^{-\beta - 1}$$

(7)

For any $p_i$ and $\lambda_i$, the second derivative of $u_i(\cdot)$ is negative, that is, $u''_i(\cdot) < 0$. Hence utility function $u_i(\cdot)$ is concave. Based on the above analysis, utility function is non-decreasing and concave.

For all $i \in \{1, 2, ..., n\}$, the objective function is nonlinear and concave with respect to $\tau_i$. The hardness of general integer programming problem is no less than NP-complete, where the objective function is linear with variables. Therefore, the hardness of finding optimal timer $\tau_i$ for each content, as formulated in (5), is NP-complete at least.

### 4 Linear Programming Based Approximation Algorithm for Content Timer

As analyzed in Section 3, the hardness of finding optimal timer for each content is no less than NP-complete. In this section, we propose a linear programming based approximation algorithm for calculating approximate optimal timer for each content. And the upper bound of approximation ratio is proved.

According to the definition of $u_i(\cdot)$, we propose piecewise linear function $h_i(x)$ as follows, where $r \in \{0, 1, ..., T - 1\}$.

$$h_i(x) = \begin{cases} (u_i(r + 1) - u_i(r))(x - r) + u_i(r), x \in (r, r + 1); \\ u_i(x), x \in \{0, 1, ..., T\} \end{cases}$$

(8)

For given $p_i$ and $\lambda_i$, $u_i(\cdot)$ is defined as follows.

$$u_i(x) = p_i \left(1 - e^{-\lambda_i x}\right)^{1-\beta}$$

(9)

**Lemma 2** The proposed piecewise linear function $h_i(x)$ is non-decreasing and concave.

**Proof** For any $r \in \{0, 1, ..., T\}$, based on the definition of $h_i(\cdot)$, if $x_1 < x_2$, there must exist $r_1, r_2 \in \{0, 1, ..., T\}$ to make $x_1 \in [r_1, r_1 + 1]$ and $x_2 \in [r_2, r_2 + 1]$ be true. If $r_1 = r_2$, then we know $x_1, x_2 \in [r_1, r_1 + 1]$, and the following can be obtained:

$$h_i(r_1) \leq h_i(x_1) < h_i(x_2) \leq h_i(r_1 + 1).$$

If $r_1 < r_2$, then it is easily known that

$$h_i(x_1) \leq h_i(r_1) \leq h_i(r_2) < h_i(x_2).$$

Thus piecewise function $h_i(\cdot)$ is non-decreasing.

The second derivative of $h_i(\cdot)$ with respect to $x$ can be derived as $h''_i(x) = 0$, when $x \in (r, r + 1)$. Then $h_i(x)$ satisfies the sufficient condition of concave function in the case of $x \in (r, r + 1)$. In the following, we prove that $h_i(x)$ is a concave function when $x \in \{0, 1, ..., T\}$. For any $z \in \{0, 1, ..., T\}$, we assume that there exist $x_1, x_2, ..., x_m \in [0, T]$ and $\lambda_1, \lambda_2, ..., \lambda_m \in [0, 1]$, such that the following equation holds.

$$\sum_{k=1}^{m} \lambda_k = 1, z = \sum_{k=1}^{m} \lambda_k x_k.$$  

(10)

According to Lemma 1, $u_i(\cdot)$ is a concave function with respect to $x$ in the case of $x \in (0, +\infty)$. Then we can have

$$u_i(z) \geq \sum_{k=1}^{n} \lambda_k u_i(x_k)$$

(11)

For $x \in [0, 1]$, there must exist $r_k$ to make $x_k \in [r_k, r_k + 1]$, and the following can be obtained:

$$h_i(x_k) = (u_i(r_k + 1) - u_i(r_k))(x_k - r_k) + u_i(r_k) = (x_k - r_k) u_i(r_k + 1) + (r_k + 1 - x_k) u_i(r_k)$$

(12)

It can be known that $x_k - r_k, r_k + 1 - x_k \in [0, 1]$ and $(x_k - r_k) + (r_k + 1 - x_k) = 1$. Based on the definition of concave function, we have the following:

$$h_i(x_k) = (x_k - r_k) u_i(r_k + 1) + (r_k + 1 - x_k) u_i(r_k) \leq u_i((x_k - r_k)(r_k + 1) + (r_k + 1 - x_k) r_k) = u_i(x_k)$$

(13)
And we known that
\[ h_i(z) = u_i(z) \geq \sum_{k=1}^{n} \lambda_k u_i(x_k) \geq \sum_{k=1}^{n} \lambda_k h_i(x_k) \]  (13)

Based on the above analysis, piecewise function \( h_i(x) \) is concave.

According to the constraints of integer optimization problem (5), we propose an optimization problem in real number as follows.

\[
\max \sum_{i=1}^{n} h_i(x_i) \\
\text{s.t.} \sum_{i=1}^{n} x_i \leq BT, \\
x_i \in [0, T], \quad i \in \{1, 2, ..., n\}
\]  (14)

**Theorem 1** The optimal objective value of problem (14) is a higher bound of that of integer optimization problem (5).

**Proof** We suppose that \( \tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n \) are the optimal solutions of integer optimization problem (5), which make the objective achieve the maximum. Thus we know that \( \tilde{x}_1, ..., \tilde{x}_n \) satisfy the constraints of problem (14), and they are the feasible solutions of this problem. We can derive the following:

\[
\max \sum_{i=1}^{n} h_i(x_i) \geq \sum_{i=1}^{n} h_i(\tilde{x}_i) = \sum_{i=1}^{n} u_i(\tilde{x}_i) = \\
\max \sum_{i=1}^{n} \frac{p_i}{1-\beta} (1-e^{-\lambda_i \tilde{x}_i})^{1-\beta}
\]  (15)

Thus the optimal objective value of problem (14) is a higher bound of that of integer optimization problem (5).

**Theorem 2** The proposed optimization problem (14) is equivalent to the following linear programming technique, and then we can solve problem (14) through linear programming technique.

\[
\max \sum_{i=1}^{n} \sum_{k=0}^{T} y_{i,k} \frac{p_i}{1-\beta} (1-e^{-\lambda_i k})^{1-\beta} \\
\text{s.t.} \sum_{i=1}^{n} \sum_{k=0}^{T} y_{i,k} k \leq BT; \\
\sum_{k=0}^{T} y_{i,k} = 1, \quad i \in \{1, 2, ..., n\}; \\
y_{i,k} \in [0, 1], \quad i \in \{1, 2, ..., n\}, \quad k \in \{0, 1, ..., T\}
\]  (16)

**Proof** It is supposed that \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n \) are the optimal solutions of problem (14). For \( \forall i \in \{1, 2, ..., n\} \), there must exist \( r_1, ..., r_n \) to make \( \hat{x}_i \in [r_i, r_i + 1] \) be true, in which \( r_i \in \{0, 1, ..., T - 1\} \). For \( \forall i \in \{1, 2, ..., n\} \), \( \hat{y}_{i,k} \) can be assigned as follows.

\[
\hat{y}_{i,k} = \begin{cases} 
    r_k + 1 - \hat{x}_i, & k = r_i; \\
    \hat{x}_i - r_i, & k = r_i + 1; \\
    0, & k \neq r_i, r_i + 1
\end{cases}
\]  (17)

It is easily known that

\[
\max \sum_{i=1}^{n} \sum_{k=0}^{T} \hat{y}_{i,k} k = \sum_{i=1}^{n} (r_i + 1 - \hat{x}_i)(\hat{x}_i - r_i)(r_i + 1) = \\
\sum_{i=1}^{n} \hat{x}_i \leq BT
\]  (18)

According to the definition of \( \hat{y}_{i,k} \), for \( \forall i \in \{1, 2, ..., n\} \), we know that \( \sum_{k=0}^{T} \hat{y}_{i,k} = 1 \) and \( \hat{y}_{i,k} \in [0, 1] \). Then \( \hat{y}_{1,0}, \hat{y}_{1,1}, ..., \hat{y}_{n,T} \) are the feasible solutions of optimization problem (16), and the following can be derived.

\[
\max \sum_{i=1}^{n} \sum_{k=0}^{T} \hat{y}_{i,k} \frac{p_i}{1-\beta} (1-e^{-\lambda_i k})^{1-\beta} \geq \\
\sum_{i=1}^{n} \sum_{k=0}^{T} \hat{y}_{i,k} \frac{p_i}{1-\beta} (1-e^{-\lambda_i k})^{1-\beta} = \\
\sum_{i=1}^{n} (r_i + 1 - \hat{x}_i) \frac{p_i}{1-\beta} (1-e^{-\lambda_i r_i})^{1-\beta} + \\
\sum_{i=1}^{n} (\hat{x}_i - r_i) \frac{p_i}{1-\beta} (1-e^{-\lambda_i (r_i+1)})^{1-\beta} = \\
\sum_{i=1}^{n} \frac{p_i}{1-\beta} ((1-e^{-\lambda_i (r_i+1)})^{1-\beta}-(1-e^{-\lambda_i r_i})^{1-\beta}) + \\
\sum_{i=1}^{n} \frac{p_i}{1-\beta} (1-e^{-\lambda_i r_i})^{1-\beta} = \\
\sum_{i=1}^{n} h_i(\hat{x}_i) = \max \sum_{i=1}^{n} h_i(x_i)
\]  (19)

Therefore, the optimal objective value of linear programming problem (16) is a higher bound of that of general optimization problem (14).

We suppose that \( \hat{y}_{1,0}, \hat{y}_{1,1}, ..., \hat{y}_{n,T} \) are the optimal solutions of linear programming problem (16). For \( \forall i \in \{1, 2, ..., n\} \), \( \hat{x}_i \) is assigned as follows.

\[
\hat{x}_i = \sum_{k=0}^{T} \hat{y}_{i,k} k.
\]

Since \( \sum_{k=0}^{T} \hat{y}_{i,k} = 1 \), we can derive
the optimal solutions of linear programming problem (14), which are the feasible solutions to the problem. According to \( h_1(\cdot) \), we have that

\[
\max_{i=1, k=0}^{n, T} \sum y_{i,k} \frac{p_i - (1 - e^{-\lambda_i k})^{1-\beta}}{1 - \beta} = \sum_{i=1, k=0}^{n, T} \hat{y}_{i,k} \cdot u_i(k) = \sum_{i=1, k=0}^{n, T} \hat{y}_{i,k} \cdot h_1(k)
\]

Based on Lemma 2, \( h_1(\cdot) \) is non-decreasing and concave. We can have that

\[
\sum_{i=1, k=0}^{n, T} \hat{y}_{i,k} \cdot h_1(k) = h_1(\sum_{i=1, k=0}^{n, T} \hat{y}_{i,k}) = h_1(\hat{x}_i).
\]

Thus the following can be known.

\[
\sum_{i=1, k=0}^{n, T} \hat{y}_{i,k} \cdot h_1(k) \leq h_1(\hat{x}_i) = \max_{i=1}^{n} h_1(x_i)
\]

According to inequalities (20) and (21), the optimal objective value of problem (14) is a higher bound of that of linear programming problem (16).

Based on the above analysis, the general problem (14) is equivalent to the linear programming problem (16). Thus we can solve problem (14) through linear programming technique.

By linear programming technique, we can obtain the optimal solutions of problem (14). And we can construct feasible solutions to problem (5) by rounding down the optimal solutions. Then the procedures of Linear Programming based Approximation algorithm for content Timer (LPAT) are easily designed. We suppose that \( \hat{y}_{1,0}, \hat{y}_{1,1}, ..., \hat{y}_{n,T} \) are the optimal solutions of linear programming problem (16). \( \forall i \in \{1, 2, ..., n\} \), let \( \hat{x}_i = \sum_{k=0}^{T} \hat{y}_{i,k} k \). According to the proof of Theorem 2, \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n \) are the optimal solutions of problem (14). Then we can get the feasible solutions of integer optimization problem (5) by rounding down \( \hat{x}_i \).

**Theorem 3** We suppose that \( \hat{y}_{1,0}, \hat{y}_{1,1}, ..., \hat{y}_{n,T} \) are the optimal solutions of linear programming problem (16). \( \forall i \in \{1, 2, ..., n\} \), let \( \hat{x}_i = \sum_{k=0}^{T} \hat{y}_{i,k} k \). Then \( [\hat{x}_1], [\hat{x}_2], ..., [\hat{x}_n] \) are the feasible solutions of integer optimization problem (5). The approximation ratio \( \gamma \) satisfies the following,

\[
\gamma < \frac{p_{\max}(1 - e^{-\lambda_{\max} T})^{1-\beta}}{p_{\min}(1 - e^{-\lambda_{\min} T})^{1-\beta}}
\]

where \( p_{\max} = \max\{p_i\}, p_{\min} = \min\{p_i\}, \lambda_{\max} = \max\{\lambda_i\}, \lambda_{\min} = \min\{\lambda_i\}, i \in \{1, 2, ..., n\} \).

**Proof** According to the proof of Theorem 2, \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n \) are the optimal solutions of problem (14). It is easily known that \( [\hat{x}_1], [\hat{x}_2], ..., [\hat{x}_n] \) satisfy the constraints of integer optimization problem (5), and they are the feasible solutions. It is supposed that \( \hat{r}_1, \hat{r}_2, ..., \hat{r}_n \) are the optimal solutions of integer optimization problem (5), which make the objective achieve the maximum. Let \( \gamma \) denote the approximation ratio, and we can derive the following:

\[
\gamma = \frac{\sum_{i=1}^{n} p_i (1 - e^{-\lambda_i t_i})^{1-\beta}}{\sum_{i=1}^{n} p_i (1 - e^{-\lambda_i \hat{x}_i})^{1-\beta}} \leq \frac{\sum_{i=1}^{n} p_i (1 - e^{-\lambda_i \hat{x}_i})^{1-\beta}}{\sum_{i=1}^{n} p_i (1 - e^{-\lambda_i \hat{x}_i})^{1-\beta}} < \frac{\sum_{i=1}^{n} p_i (1 - e^{-\lambda_i \hat{x}_i})^{1-\beta}}{\sum_{i=1}^{n} p_i (1 - e^{-\lambda_i \hat{x}_i})^{1-\beta}} < \frac{p_{\max}(1 - e^{-\lambda_{\max} T})^{1-\beta}}{p_{\min}(1 - e^{-\lambda_{\min} T})^{1-\beta}}
\]

in which, \( p_{\max} = \max\{p_i\}, p_{\min} = \min\{p_i\}, \lambda_{\max} = \max\{\lambda_i\}, \lambda_{\min} = \min\{\lambda_i\}, i \in \{1, 2, ..., n\} \).

When the approximate optimal timers are obtained, a simple schedule strategy can be designed as follows.

If a content miss happens, the cached content with the least timer is replaced by the new request one.

If the timer of a cached content becomes zero, then it is replaced by the one with the largest timer.

5 Lagrange Multiplier Based Algorithm for Content Timer with Relaxed Constraints

Linear programming based approximation algorithm can provide content timer with approximation guarantee. However, the quantity of available contents
often achieves millions and linear programming based algorithm incurs high computation overhead. In practical applications, content providers hold large amount of contents requested by end users, but some contents have similar popularity in a period of time, such as hot news and popular videos. For \( \forall i \in \{1, 2, ..., n\} \), utility function \( u_i(\cdot) \) is non-decreasing and concave, and the increasing is reduced with \( \tau_i \). There exists high diversity among popular contents, thus one content can not be kept all the time in view of utility gain. Based on the above analysis, we propose content caching with relaxed constraints as follows, which loosens the range of \( \tau_i \).

\[
\max \sum_{i=1}^{n} p_i \frac{(1 - e^{-\lambda_i x_i})^{1-\beta}}{1 - \beta} \quad \text{s.t.} \sum_{i=1}^{n} \tau_i \leq BT, \quad \tau_i \in Z, \forall i \in \{1, 2, ..., n\}
\]

(24)

Similar analysis as in Section 3.2, the hardness of integer optimization problem (24) is NP-complete at least. Similarly, we propose content caching with relaxed constraints in real numbers as follows.

\[
\max \sum_{i=1}^{n} p_i \frac{(1 - e^{-\lambda_i x_i})^{1-\beta}}{1 - \beta} \quad \text{s.t.} \sum_{i=1}^{n} x_i \leq BT, \quad x_i \in R, \forall i \in \{1, 2, ..., n\}
\]

(25)

Theorem 4 If \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n, \hat{\sigma} \) are the solutions of the following equations, then \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n \) are the optimal solutions of optimization problem (25), which makes the objective achieve the maximum.

\[
\begin{align*}
\frac{\partial L}{\partial x_1} &= p_1 \lambda_1 e^{-\lambda_1 x_1} (1 - e^{-\lambda_1 x_1})^{-\beta} - \sigma = 0; \\
&\vdots \\
\frac{\partial L}{\partial x_n} &= p_n \lambda_n e^{-\lambda_n x_n} (1 - e^{-\lambda_n x_n})^{-\beta} - \sigma = 0; \\
\frac{\partial L}{\partial \sigma} &= \sum_{i=1}^{n} x_i - BT = 0
\end{align*}
\]

(26)

Proof \( \forall i \in \{1, 2, ..., n\} \), we know that \( \frac{p_i}{1 - \beta} (1 - e^{-\lambda_i x_i})^{1-\beta} \) is increasing with respect to \( x \). Then the objective achieves the maximum, if and only if the following equation holds.

\[
\sum_{i=1}^{n} x_i = BT.
\]

Thus, we can obtain the optimal solutions to problem (25) by Lagrange multiplier method. We define Lagrange function as follows.

\[
L(x_1, ..., x_n, \sigma) = \sum_{i=1}^{n} p_i \frac{1 - e^{-\lambda_i x_i}}{1 - \beta} - \sigma \left( \sum_{i=1}^{n} x_i - BT \right)
\]

And the following equation can be derived.

\[
\frac{\partial L}{\partial x_i} = p_i \lambda_i e^{-\lambda_i x_i} (1 - e^{-\lambda_i x_i})^{-\beta} - \sigma
\]

(27)

It is easily known that

\[
\frac{\partial^2 L}{\partial x_i \partial x_j} = 0, i \neq j;
\]

\[
\frac{\partial^2 L}{\partial x_i^2} = -p_i \lambda_i^2 e^{-\lambda_i x_i} (1 - e^{-\lambda_i x_i})^{-\beta} - \beta p_i \lambda_i e^{-\lambda_i x_i} (1 - e^{-\lambda_i x_i})^{-\beta} - 1
\]

(28)

Hence, Lagrange function is decreasing with respect to \( x_i \), and the Hessian matrix of \( L(\cdot) \) is shown as Eq. (29)

\[
\frac{\partial^2 L}{\partial x_i \partial x_j} = \begin{pmatrix} \frac{\partial^2 L}{\partial x_i^2} & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\partial^2 L}{\partial x_n \partial \sigma} \end{pmatrix}
\]

(29)

It is easily known that all the eigenvalues of matrix (29) are negative. Thus the matrix is negative definite. In conclusion, if \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n, \hat{\sigma} \) are the solutions of Eqs. (26), then \( \hat{x}_1, \hat{x}_2, ..., \hat{x}_n \) are the optimal solutions of optimization problem (25), which makes the objective achieve the maximum.

It is assumed that \( \hat{\sigma} \) is the solution of Eqs. (26). \( \forall i \in \{1, 2, ..., n\} \), the optimal solution of problem (25), denoted as \( \hat{x}_i \), satisfies the following equation.

\[
\frac{p_i \lambda_i e^{-\lambda_i \hat{x}_i}}{(1 - e^{-\lambda_i \hat{x}_i})^\beta} = \hat{\sigma}.
\]

Thus, it is hard to provide the analytic expressions of \( \hat{x}_i \). We design approximation algorithm with polynomial time complexity for feasible solutions of problem (24). According to Eq. (28), \( p_i \lambda_i e^{-\lambda_i x_i} (1 - e^{-\lambda_i x_i})^{-\beta} \) is decreasing with \( x_i \). For \( \forall i \in \{1, 2, ..., n\} \), \( \sigma \) satisfies the following:

\[
0 < \sigma \leq p_i \lambda_i.
\]

Hence, the upper bound of \( \sigma \) is

\[
\min_{i \in \{1, ..., n\}} \{p_i \lambda_i\}.
\]

For given \( i \) and \( \hat{\sigma} \), we assume \( \hat{x}_i \) is the solution of the following equation.

\[
\frac{e^{-\lambda_i x_i}}{(1 - e^{-\lambda_i x_i})^\beta} = \frac{\hat{\sigma}}{p_i \lambda_i}
\]

(30)
Then we have that

\[ e^{-\lambda_i \tilde{x}_i} \leq \frac{\tilde{\sigma}}{p_i \lambda_i} \leq \frac{e^{-\lambda_i \tilde{x}_i}}{1 - e^{-\lambda_i \tilde{x}_i}}. \]

The following can be derived.

\[ \frac{\tilde{\sigma}}{p_i \lambda_i} \leq e^{-\lambda_i \tilde{x}_i} \leq \frac{\tilde{\sigma}}{p_i \lambda_i} - \ln \tilde{\sigma}. \]

The solutions of problem (24) restrict to integers. Then for given \( \tilde{\sigma}, \tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n \) are obtained based on Eq. (31). To get feasible solutions of optimization problem (24), \( \tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n \) are scaled as follows.

Firstly, we compute scaling coefficient, denoted as \( \kappa \).

\[ \kappa = BT \sum_{i=1}^{n} \tilde{x}_i. \]

Secondly, \( \forall i \in \{1, 2, ..., n\} \), \( \tilde{r}_{i, \tilde{\sigma}} \) is assigned as formula (32).

\[ \tilde{r}_{i, \tilde{\sigma}} = \left\lfloor \kappa \tilde{x}_i \right\rfloor \] (32)

We easily know that \( \sum_{i=1}^{n} \kappa \tilde{x}_i = BT \). Thus \( \tilde{r}_{1, \tilde{\sigma}}, \tilde{r}_{2, \tilde{\sigma}}, ..., \tilde{r}_{n, \tilde{\sigma}} \) are the feasible solutions of problem (24).

Based on the above analysis, the approximate solutions can be obtained. We get approximation of \( \sigma \), which can be calculated as follows.

\[ \tilde{\sigma} = \arg \max \left\{ \sum_{i=1}^{n} p_i \left( 1 - e^{-\lambda_i \tilde{r}_{i, \tilde{\sigma}}} \right)^{1-\beta} \right\}. \]

And then feasible solutions are returned, which are obtained based on Eq. (32).

The pseudo-code of Lagrange Multiplier based Approximation algorithm for content Timer (LMAT) is presented in Algorithm 1. It is easily known that the algorithm yields time computation of \( O(K \sigma_{\text{min}} n T) \), where \( k \) is given at first and \( \sigma_{\text{min}} = \min_{i \in \{1, 2, ..., n\}} \{ p_i \lambda_i \} \). Thus the time complexity of the algorithm is \( O(K T n \min_{i \in \{1, 2, ..., n\}} \{ p_i \lambda_i \}) \).

### 6 Experiment Evaluation

In this section, we evaluate the effectiveness of the proposed algorithms through numerical simulations. Several experiments are carried out to show the relationships between the capacity and the performance of caching, such as hit rate, utility gain, and approximation ratio. The simulations are implemented by Matlab.

We used the trace of video request derived from the YouTube dataset\(^{[28]}\) for the evaluations, which provides about \( 5 \times 10^7 \) requests for the available videos. The simulation of the content request process is as follows. Each request for the video is considered as content requesting, and we assume each video is with the same size. The comparisons with the benchmark schemes are conducted.

**LRU:** The cache node maintains a list of cached contents, which are ordered by the latest time of the content caching in the node. When the node is full, the least recently requested content is replaced by the request content, which is not cached in the node.

**LFU:** Similarly to LRU, the cache node maintains a list of cached content, which are ordered by the number of request among all the content. When the node is full, the content with least request number is replaced by the new content, which is not cached in the node.

The first group of experiments is to investigate the hit rate of the proposed algorithms. Figure 1 shows
the relationship between the capacity and hit rate. The proposed algorithms have better performance. For example, the hit rate is 38%, when the capacity is 6000, which is 1.2% to the total content. The metrics of the benchmark schemes are based on the current information of request content. If a content is replaced by a new one, then the request history of the content is missed. The utility function considers the popularity and request process, which is helpful for the future decision. The popularity has effect on the content timer, then accurate estimation of popularity is in favor of content hit.

The second group of experiments is to investigate the utility gain of the proposed algorithms, when the capacity varies in the range of $[0, 2 \times 10^5]$. Let $r_i$ denote the hit rate of content $i$. The utility gain is the sum of utility of content $i$, if content $i$ is cached on the node. Utility gain can be calculated as

$$\sum 2p_i \sqrt{r_i}.$$  

Figure 2 shows the comparison of utility gain among LPAT, LMAT, and benchmark algorithms. As expected, the proposed algorithms can reach higher utility gain than that of benchmark algorithms. The utility function considers the popularity and the fairness, which makes the proposed algorithm achieve higher hit rate. Therefore, our algorithms can improve the caching efficiency.

The third group of experiments is to investigate the computing performance of the proposed algorithms and the correctness of the approximation ratio is verified. According to Theorem 2, the objective function value of problem (16) is a higher bound of that of problem (5). In the experiments, approximation ratio is the ratio of the objective value of problem (16) to that of approximate solutions returned by LPAT and LMAT algorithms. Figure 3 depicts the approximation ratio obtained by the proposed algorithms. From the experimental results, we can infer that the proposed approximation algorithms can achieve high accuracy in terms of content timers.

7 Conclusion

Content caching is critical to the optimization of network architecture. The problem of finding optimal timer for each content is defined, and is formalized as an integer optimization problem with the objective of maximizing aggregate utility. In view of the hardness of this problem, a linear programming based approximation algorithm is proposed, and the approximation ratio is proved. Furthermore, the problem of content caching with relaxed constraints is given. A Lagrange multiplier based approximation algorithm with polynomial time complexity is provided. Simulation results show that the proposed algorithms have better performance for utility gain.

A good direction for future work would be to study weights of the utility function and optimize the cache scheduling strategy based on the timer and...
content popularity. We also hope to construct a utility-based content caching framework for heterogeneous networks.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Nos. 61572104 and 61402076), Startup Fund for the Doctoral Program of Liaoning Province (No. 20141023), the Fundamental Research Funds for the Central Universities (Nos. DUT15RC(3)088, DUT15QY26, and DUT14QY06).

References

[1] X. Zheng, Z. Cai, J. Li, and H. Gao, Scheduling flows with multiple service frequency constraints, *IEEE Internet of Things*, 2016, doi: 10.1109/JIOT.2016.2577630.

[2] J. Li, Z. Cai, M. Yan, and Y. Li, Using crowdsourced data in location-based social networks to explore influence maximization, in *Proc. 2016 IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, 2016.

[3] Z. Duan, M. Yan, Z. Cai, X. Wang, M. Han, and Y. Li, Truthful incentive mechanisms for social cost minimization in mobile crowdsourcing systems, *Sensors*, vol. 16, no. 4, pp. 481–493, 2016.

[4] X. Zheng, Z. Cai, J. Li, and H. Gao, An application-aware scheduling policy for real-time traffic, in *Proc. 35th International Conference on Distributed Computing Systems (ICDCS)*, Columbus, OH, USA, 2015, pp. 421–430.

[5] S. Li, J. Xu, M. D. Schaaf, and W. Li, Popularity-driven content caching, in *Proc. 2016 IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, 2016.

[6] Amazon CloudFront, http://aws.amazon.com/cloudfront/details/, 2016.

[7] E. Nygren, R. K. Sitaraman, and J. Sun, The akamai network: A platform for high-performance internet applications, *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[8] L. Muscariello, G. Carofiglio, and M. Gallo, Bandwidth and storage sharing performance in information centric networking, in *Proc. 2011 SIGCOMM Workshop on Information-Centric Networking*, Toronto, ON, Canada, 2011, pp. 26–31.

[9] B. Tan and L. Massoulié, Optimal content placement for peer-to-peer video-on-demand systems, *Transactions on Networking (TON)*, vol. 21, no. 2, pp. 566–579, 2013.

[10] M. Dehghan, L. Massoulié, D. Towsley, D. Menasche, and Y. C. Tay, A utility optimization approach to network cache design, in *Proc. 2016 IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, 2016.

[11] B. Ahlgren, C. Dannenwitz, C. Imbrenda, D. Kutscher, and B. Ohlman, A survey of information-centric networking, *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, and R. Braynard, Networking named content, *Communications of the ACM*, vol. 55, no. 1, pp. 117–124, 2012.

[13] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and P. Lau, Scaling social media applications into geo-distributed clouds, *Transactions on Networking*, vol. 23, no. 3, pp. 689–702, 2015.

[14] P. Blasco and D. Gunduz, Learning-based optimization of cache content in a small cell base station, in *Proc. 2014 IEEE International Conference on Communications (ICC)*, Sydney, Australia, 2014, pp. 1897–1903.

[15] J. Famaey, F. Ierbeke, T. Wauters, and S. De Turck, Towards a predictive cache replacement strategy for multimedia content, *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 219–227, 2013.

[16] S. Guo, H. Xie, and G. Shi, Collaborative forwarding and caching in content centric networks, in *Proc. 11th International IFIP TC 6 Networking Conference*, Prague, Czech Republic, 2012, pp. 41–55.

[17] L. Zhang, X. Wang, J. Lu, P. Li, and Z. Cai, An efficient privacy preserving data aggregation approach for mobile sensing, *Security and Communication Networks Journal*, 2016, doi: 10.1002/sec.1546.

[18] L. Zhang, Z. Cai, and X. Wang, FakeMask: A novel privacy preserving approach for smartphones, *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[19] M. A. Maddah-Ali and U. Niesen, Fundamental limits of caching, *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[20] J. Hachem, N. Karamchandani, and S. Diggavi, Content caching and delivery over heterogeneous wireless networks, in *Proc. 2015 IEEE Conference on Computer Communications (INFOCOM)*, Hong Kong, China, 2015, pp. 756–764.

[21] M. S. ElBamby, M. Bennis, W. Saad, and M. Latvaaho, Content-aware user clustering and caching in wireless small cell networks, in *11th International Symposium on Wireless Communications Systems (ISWCS)*, Barcelona, Spain, 2014, pp. 945–949.

[22] M. Lecoute, G. Paschos, L. Gkatzikis, M. Draief, S. Vassiliras, and S. Chouvardas, Placing dynamic content in caches with small population, in *Proc. 2016 IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, 2016.

[23] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. K. Sitaraman, On the complexity of optimal routing and content caching in heterogeneous networks, in *Proc. 2015 IEEE Conference on Computer Communications (INFOCOM)*, Hong Kong, China, 2015, pp. 936–944.

[24] F. P. Kelly, A. K. Maulloo, and D. K. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[25] L. Huang and M. J. Neely, Utility optimal scheduling in energy harvesting networks, *Transactions on Networking*, vol. 21, no. 4, pp. 1117–1130, 2013.
Ran Bi received the PhD, MS, and BS degrees from Harbin Institute of Technology, China in 2015, 2010, and 2008, respectively. Dr. Bi is currently a lecturer with the School of Computer Science and Technology, Dalian University of Technology, China. Her research interests include wireless networking, sensory data management, Internet of Things, and vehicular networking.

Xu Zheng received the BS and MS degrees from Harbin Institute of Technology in 2010 and 2012, respectively, where he is pursuing his PhD degree. Mr. Zheng is currently a visiting PhD student in the Department of Computer Science at Georgia State University. His research interest is wireless network.

Yingshu Li received the PhD and MS degrees from University of Minnesota-Twin Cities in 2005 and 2003, respectively. She received the BS degree from Beijing Institute of Technology, China in 2001. Dr. Li is currently an associate professor in the Department of Computer Science at Georgia State University. Her research interests include wireless networking, sensor networks, sensory data management, social networks, and optimization. Her research has been supported by the National Science Foundation (NSF) of USA, the National Science Foundation of China (NSFC), the Electronics and Telecommunications Research Institute (ETRI) of South Korea, and GSU internal grants. Dr. Li is the recipient of an NSF CAREER Award.