RecCac: Recommendation-Empowered Cooperative Edge Caching for Internet of Things

HAN Suning[1], LI Xiuhua[1], SUN Chuan[1], WANG Xiaofei[2], Victor C. M. LEUNG [3,4]

(1. Chongqing University, Chongqing 400000, China;
2. Tianjin University, Tianjin 300072, China;
3. Shenzhen University, Shenzhen 518000, China;
4. The University of British Columbia, Vancouver V6T 1Z4, Canada)

Abstract: Edge caching is an emerging technology for supporting massive content access in mobile edge networks to address rapidly growing Internet of Things (IoT) services and content applications. However, the edge server is limited with the computation/storage capacity, which causes a low cache hit. Cooperative edge caching jointing neighbor edge servers is regarded as a promising technique to improve cache hit and reduce congestion of the networks. Further, recommender systems can provide personalized content services to meet user's requirements in the entertainment-oriented mobile networks. Therefore, we investigate the issue of joint cooperative edge caching and recommender systems to achieve additional cache gains by the soft caching framework. To measure the cache profits, the optimization problem is formulated as a 0 - 1 Integer Linear Programming (ILP), which is NP-hard. Specifically, the method of processing content requests is defined as server actions, we determine the server actions to maximize the quality of experience (QoE). We propose a cache-friendly heuristic algorithm to solve it. Simulation results demonstrate that the proposed framework has superior performance in improving the QoE.

Keywords: IoT; recommender systems; cooperative edge caching; soft caching

DOI: 10.12142/ZTECOM.202102002

https://kns.cnki.net/kcms/detail/34.1294.TN.20210525.1320.002.html, published online May 26, 2021

Manuscript received: 2021-03-01

Citation (IEEE Format): S. N. Han, X. H. Li, C. Sun, et al. "RecCac: recommendation-empowered cooperative edge caching for Internet of Things," *ZTE Communications*, vol. 19, no. 2, pp. 02 - 10, Jun. 2021. doi: 10.12142/ZTECOM.202102002.

1 Introduction

As the development trend of future networks, the Internet of things (IoT) has become a hot research topic in the industry and academia in recent years[1]. The emergence of "IoT" paradigm makes accessibility of various IoT sensors (e.g., smart cameras and temperature sensors) universal, and thus enables intelligent services to improve the life quality of humans[2]. Billions of IoT devices (IDs) generate a tremendous number of monitoring data while a great many end-users are consuming these data. However, countless electronic devices are anticipated to generate a sheer volume of traffic loads, and the aggregate load on core networks is expected to be large. Therefore, it is important to reduce congestion and transmission delay for network providers[3 - 5].

As we have stated above, mobile edge networks are faced with the challenge of the explosive growth of IoT data requests from the IDs, especially in the current backhaul net-

This work is supported in part by National Key R&D Program of China under Grant Nos. 2018YFB2100100 and 2018YFF0214700, National NSFC under Grant Nos. 61902044 and 62072060, Chongqing Research Program of Basic Research and Frontier Technology under Grant No. CSTC2019-jcyjmsxmX0589, Key Research Program of Chongqing Science and Technology Commission under Grant Nos. CSTC2017jcyjBX0025 and CSTC2019jscx-zdztzxX0031, Fundamental Research Funds for the Central Universities under Grant No. 2020CDJQY-A022, Chinese National Engineering Laboratory for Big Data System Computing Technology, and Canadian NSERC.

works[6]. According to the research, most of the high load in the mobile networks is generated by downloading the same content and data. To solve this problem, it is necessary to put forward new revolutionary methods in network structure and data transmission[7]. As one of the rapidly developing technologies, edge caching has drawn growing attention. Edge caching technology can reduce repeated downloading and transmission by caching contents in advance[8]. However, as content providers (CPs) provide growing content, and the storage and computing capacity of a cell (e.g., edge server) are limited, we still face great challenges to solve the above problems. Many researchers are looking for additional cache gains in this area. Some current research (e. g., FemtoCache[9]) focuses on caching contents in the edge server of base stations (BSs). However, it only focuses on the basic cell cache, and the understanding of inter-cell cooperation is not deep.

Besides, how to use the cached contents to achieve more cache gains is also a problem we have to consider. It is difficult to improve the caching performance only by focusing on the content popularity in the entertainment-oriented mobile networks. To solve this problem, the recommender systems provide an effective method that can provide personalized content recommendations through historical behavior, e.g., users may have evaluated or scored different contents. However, some related content, such as two similar comedy movies or two short videos of the same type, might have similar utility for a user. We use the term—soft caching[10], which means that if the local BS doesn't cache the requested content, the BS can send other relevant contents available locally. If the user likes or accepts the relevant contents (under a certain threshold) instead of the content which was originally requested, a soft cache hit will occur. This scheme may give up some content relevance, but it avoids the "expensive" connection of the IDs to get the requested content from the backhaul network. Actually, some recent experimental evidence suggests that IDs may be willing to trade off some content relevance for a better quality of experience (QoE)[11].

More specifically in this paper, the cooperative edge caching and recommender systems are used to alleviate the pressure of the backhaul network and get related contents to achieve soft caching, respectively. We combine cooperative edge caching with recommender systems to improve the QoE. Recently, some researchers consider the interaction between edge caching and recommender systems to optimize cache or recommender systems[10-17]. However, most of the research only focuses on one side of the problem, e. g., caching-friendly recommendations[10, 12-13, 15, 17] or recommendation-aware caching policies[16]. The real joint treatment of both is tried in Refs. [11] and [14], but their studies on hierarchical mobile edge networks are not deep enough.

To sum up, different from the existing studies on edge caching and recommender systems, we focus on improving the QoE by judiciously selecting server actions. Our main contributions are summarized as follows:

1) We combine cooperative edge caching with soft caching for IoT systems. To measure cache profits, we propose a generic metric of QoE that depends on the quality of service (QoS) and the quality of recommendation (QoR).

2) We formulate the problem of optimally choosing the server actions towards maximizing the QoE. While such joint caching and recommendation problems have been proved to be NP-hard, we have proposed a cache-friendly hierarchical heuristic algorithm.

3) Trace-driven evaluation results demonstrate that our proposed scheme has superior performance on improving the cache hits and QoE finally.

The remainder of this paper is organized as follows. Section 2 discusses the proposed hierarchical cooperative edge caching model and formulates the optimization problem. Section 3 introduces a cache-friendly hierarchical heuristic algorithm to solve the problem. Section 4 evaluates the performance of the proposed framework and Section 5 concludes this paper.

2 System Model and Problem Formulation

In this section, we introduce the system model of edge caching. Specifically, we present the hierarchical cooperative edge caching architecture and topology in Section 2.1. Section 2.2 introduces the recommendation-aware content request processing model. Then we propose a QoE model, considering delay and recommendation in Section 2.3. Finally, Section 2.4 gives the problem formulation. Some key parameters are listed in Table 1.

▼**Table 1. Key Parameters**

Notation	Definition
N	Number of BSs
M	Number of IDs
F	Total number of contents
C	Cache size of a BS
D_f	Size of the content f
p_f	The content f requested probability
$a_{m,n}$	The association of the ID and the BS
$s_{n,f}$	The cache state of the content f in the BS n
$w_{m,f}$	Rating (i.e., preference) for the content f of the ID m
$v_{m,n}$	The wireless transmission rate between the ID and the BS
$\pi_{m,n,f}$	System action
$d^L_{m,n,f}$, d^E_f, d^C_f	Transmission delay of the content f between the BS n and the ID m, BS and BS, BS and CS, respectively
$r^L_{m,n,f}$, $r^E_{m,n,f}$, $r^C_{m,f}$	Content satisfaction in different server actions

BS: base station CS: cloud server

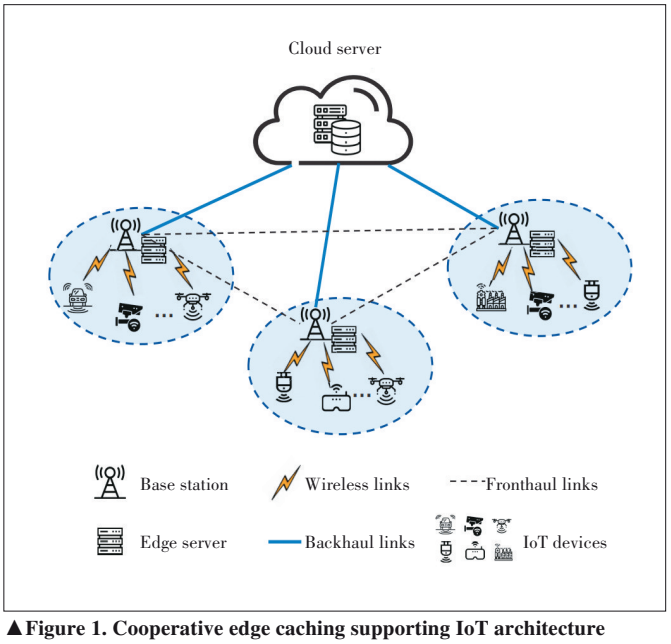
2.1 Hierarchical Cooperative Edge Caching Model

The proposed system is a cooperative Cloud-Edge-End computing system with a cloud server (CS), some discrete BSs, and IDs. As shown in Fig. 1, we consider a cooperative edge caching scenario for IoT networks. The CS has enough computing and caching capacity, consisting of all data and contents. Each BS is equipped with an edge server, which has the limited ability to cache and compute. Each ID as a content requester generates a request at each time slot. In our proposed system, each BS communicates with the CS through the backhaul links. To enhance the usage of the BSs and alleviate the pressure of the backhaul networks, each BS can communicate with all cooperative BSs through fronthaul links instead of working individually[18]. Besides, as the contents are cached in the BSs or the CS, IDs can fetch their requested contents either from edge servers via wireless links or directly by downloading the contents from the CS to the BSs.

The proposed system consisting of $\mathcal{N} = \{1, 2, ..., N\}$ fully connected BSs with a finite cache size C and $\mathcal{M} = \{1, 2, ..., M\}$ IDs are distributed in the service area of the BSs. In addition, we denote $a_{m,n} \in [0, 1]$ as the association probability between the BS n and the ID m. We assume each ID requests content or a set of data from a catalogue $\mathcal{F} = \{1, 2, ..., F\}$ at each time slot, and we denote the size of each content as D_f.

We assume that the ID m requests the content f with the standard content probability p_f^m. Hence, we could obtain the content popularity p_f from p_f^m [9]. Furthermore, we assume that the content popularity p_f changes slowly, and $\sum_{f=1}^{F} p_f = 1$.

For the cache state, we focus on whether the content has been cached in the BSs. The content cache state is denoted as $s_{n,f} \in \{0, 1\}$, $\forall n \in$, $\forall f \in F$. Here, $s_{n,f} = 1$ represents that the BS n has cached the content f, otherwise $s_{n,f} = 0$.



▲Figure 1. Cooperative edge caching supporting IoT architecture

2.2 Recommendation-Aware Request Processing Model

We define a score $w_{m,f}$ to represent the ID's preference for the content or data f. As for p_f, it denotes the probability of the ID m requesting the content f. Specifically, given the scores $w_{m,f}$, a reasonable choice could be their normalized values:

$$p_f = \frac{w_{m,f}}{\sum_{i \in \mathcal{F}} w_{m,i}}. \tag{1}$$

Since soft caching is to replace the requested content with related contents or data available in the local BS, we rank the scores in a descending order to get a recommendation list K_m of the ID m. When a content request f generated by the ID m arrives at the local BS, there are three types of situation:

1) Local hits: Local hits denote that the local BS processes content requests. The local hits are divided into direct cache hits and soft cache hits.

2) Neighboring hits: the request generated by an ID can be obtained from its cooperative BSs, and the transmission delay is relatively small compared with downloading from the CS.

3) CS hits: The ID obtains the requested content from the CS. The transmission in this situation is known as "expensive".

We model the server actions of the content request with three sub-decisions models, denoted as $\boldsymbol{\pi}_{m,n,f} = \left(\pi^L_{m,n,f}, \pi^E_{m,n,f}, \pi^C_{m,n,f}\right)$, where $\pi^L_{m,n,f}, \pi^E_{m,n,f}, \pi^C_{m,n,f} \in \{0,1\}$ are the indicators for whether the request is processed in the local BS, cooperative BSs, or the CS. Three sub-decisions can jointly determine how the request is processed. Different decisions will affect the transmission delay and content satisfaction.

As the content is indivisible, so for $\forall m \in \mathcal{M}$, only one of $\pi^L_{m,n,f}$, $\pi^E_{m,n,f}$ and $\pi^C_{m,n,f}$ can be 1. Similar to Ref. [19], the decision variable $\boldsymbol{\pi}_{m,n,f}$ is constrained by

$$\pi^L_{m,n,f} + \pi^E_{m,n,f} + \pi^C_{m,n,f} = 1. \tag{2}$$

2.3 QoE Model

We define the QoE as a combination of the QoS and the QoR. The QoS and QoR are measured by the transmission delay and content satisfaction, respectively. In the following, we will discuss the two parts with different decisions in detail:

1) Delay: We consider the transmission delay as the time for an ID to receive the contents or data. In the proposed system, there are three delay parts: $d^L_{m,n,f}$ denotes the transmission delay that the ID m receives the content from the local BS n, d^E_f denotes the transmission delay of the BSs' co-

operation, and d_f^C denotes the transmission delay between the BS and the CS.

Specifically, we assume that the wireless channel has been deployed. Similar to Ref. [20], we can get the transmission rate between the ID m and the local BS n as follow:

$$v_{m,n} = B\log_2\left(1 + \frac{P_m g_{m,n}}{\sigma^2}\right), \tag{3}$$

where B denotes the channel bandwidth; σ^2 denotes the background noise power; P_m denotes the power consumption of the BS n transmission to the ID m. The channel gain $g_{m,n}$ is estimated by the distance $l_{m,n}$ between the local BS n and the ID m.

Thus, the delay of transferring the content m between the ID f and the local BS n is denoted as:

$$d_{m,n,f}^L = a_{m,n}s_{n,f}\frac{D_f}{v_{m,n}}. \tag{4}$$

The transmission among the cooperative BSs is through fronthaul links with high bandwidth. In terms of the transmission between the CS and the BSs, the CS is usually deployed at a further distance, and a large amount of traffic is transmitted through multiple intermediate nodes; We express these two parts in terms of the average rate; v_e denotes the average transmission rate between two BSs. Therefore, the transmission delay between cooperative BSs can be expressed as follow:

$$d_f^E = \frac{D_f}{v_e}. \tag{5}$$

Similarly, v_c denotes the average transmission rate between the BSs and the CS. The transmission delay between the BSs and the CS can be expressed as:

$$d_f^C = \frac{D_f}{v_c}. \tag{6}$$

2) Recommendation: If the content requested by the ID is not cached locally, the similar contents cached locally could be alternated.

Specifically, for local hits, considering the soft caching, we define the content satisfaction as:

$$r_{m,n,f}^L = a_{m,n}s_{n,f}w_{m,f}. \tag{7}$$

Similarly, for neighboring BSs cache hits, we define the content satisfaction as:

$$r_{m,n,f}^E = s_{n,f}w_{m,f}. \tag{8}$$

For downloading the content f from the CS, we define the content satisfaction as:

$$r_{m,f}^C = w_{m,f}. \tag{9}$$

2.4 Problem Formulation

In the proposed system, our goal is to find the best server actions to improve the QoE. As we have discussed above, transmission delay and content satisfaction are major factors. We express these two parts as follows:

$$d_{m,n,f} = \frac{\pi_{m,n,f}^L}{d_{m,n,f}^L} + \frac{\pi_{m,n,f}^E}{d_{m,n,f}^L + d_f^E} + \frac{\pi_{m,n,f}^C}{d_{m,n,f}^L + d_f^C}, \tag{10}$$

$$r_{m,n,f} = r_{m,n,f}^L\pi_{m,n,f}^L + r_{m,n,f}^E\pi_{m,n,f}^E + r_{m,f}^C\pi_{m,n,f}^C, \tag{11}$$

where Eq. (10) denotes the QoS, which is expressed as the reciprocal of the delay of content transmission (i.e., when the delay of the content transmission is small, the larger QoS can be obtained), $\left(d_f^E + d_{m,n,f}^L\right)$ denotes the transmission delay when the content is sent through the cooperative BSs, and $\left(d_f^C + d_{m,n,f}^L\right)$ denotes the transmission delay when the content is downloaded from the CS. Eq. (11) denotes the QoR.

To improve the QoE, we need to trade off the QoS and the QoR (i.e., find the balance between low transmission delay and high content satisfaction) by optimizing the server actions $\pi_{m,n,f}$. To maximize the QoE, we formulate the optimization problem as:

$$\mathcal{P}: \max_{\Pi} \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}f\in\mathcal{F}}\sum p_f\left(\alpha d_{m,n,f} + \beta r_{m,n,f}\right) \tag{12a}$$

$$\text{s.t.} \quad \alpha + \beta = 1, \tag{12b}$$

$$s_{n,f} \in \{0,1\}, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{12c}$$

$$\forall\pi_{\mathrm{m,n,f}}^{\mathrm{L}} \in \{0,1\}, \forall\pi_{m,n,f}^E \in \{0,1\}, \forall\pi_{m,n,f}^C \in \{0,1\}, \tag{12d}$$

$$\sum_{n\in\mathcal{N}f\in\mathcal{F}}\sum \pi_{m,n,f} = 1, \forall m \in \mathcal{M}, \tag{12e}$$

$$\sum_{f\in\mathcal{F}} \pi_{m,n,f}D_f \leqslant \mathrm{C}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{12f}$$

where p_f denotes the probability of the content or data f requested. In Eq. (12b), α and β are the scalar parameters to balance transmission delay and content satisfaction. Eq. (12c) denotes the cache state. Eqs. (12d) and (12e) denote the constraints of the server actions. Eq. (12f) denotes the cache ability.

To represent the server actions of the system, we denote $\Pi = \left(\Pi^L, \Pi^E, ..., \Pi^C\right)$ as the entire selection, where $\Pi^L = \left(\Pi_1^L, \Pi_2^L, ..., \Pi_M^L\right)$, $\Pi^E = \left(\Pi_1^E, \Pi_2^E, ..., \Pi_M^E\right)$, and $\Pi^C = \left(\Pi_1^C, \Pi_2^C, ..., \Pi_M^C\right)$. And we denote $\Pi_m^L = \left\{\pi_{m,n,f}^L | \forall n \in \mathcal{N}, \forall f \in \mathcal{F}\right\}$,

$\Pi_m^E = \left\{\pi_{m,n,f}^E | \forall n \in \mathcal{N}, \forall f \in \mathcal{F}\right\}$, and $\Pi_m^C = \left\{\pi_{m,n,f}^C | \forall n \in \mathcal{N}, \forall f \in \mathcal{F}\right\}$.

Lemma 1: The QoE problem is equivalent to the 0 – 1 Integer Linear Programming (ILP) problem.

Proof: As mentioned above, different server actions will affect the transmission delay and content satisfaction. We denote positive constants $\mathrm{A}_{m,n,f}^1 = \alpha / \dfrac{a_{m,n} s_{n,f} D_f}{\log_2\left(1 + \dfrac{P_m g_{m,n}}{\sigma^2}\right)} + \beta a_{m,n} s_{n,f} w_{m,f}$, $\mathrm{A}_{m,n,f}^2 = \left[\alpha / \left(\dfrac{a_{m,n} s_{n,f} D_f}{\log_2\left(1 + \dfrac{P_m g_{m,n}}{\sigma^2}\right)} + \dfrac{D_f}{v_e}\right)\right] + \beta s_{n,f} w_{m,f}$, and $\mathrm{A}_{m,n,f}^3 = \left[\alpha / \left(\dfrac{a_{m,n} s_{n,f} D_f}{\log_2\left(1 + \dfrac{P_m g_{m,n}}{\sigma^2}\right)} + \dfrac{D_f}{v_c}\right)\right] + \beta w_{m,f}$. To combine optimization objectives with decision variables, the optimization objective of the problem in Eq. (12) can be expressed as:

$$\mathcal{P}: \max_{\boldsymbol{\Pi}} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}} \mathrm{p_f}\left(\mathrm{A}_{m,n,f}^1 \pi_{m,n,f}^L + \mathrm{A}_{m,n,f}^2 \pi_{m,n,f}^E + \mathrm{A}_{m,n,f}^3 \pi_{m,n,f}^C\right), \tag{13a}$$

$$\text{s.t. the same as Eq.s (12b), (12c), (12d), (12e), and (12f).} \tag{13b}$$

Thus, the problem can be described as selecting optimal server actions for processing requests with jointing transmission delay and content satisfaction. This is a 0 – 1 ILP problem, which is NP-hard. Because the number of IDs, BSs, and contents can be large, it is of high complexity to get the optimal solution by using exact methods.

3 Proposed Framework Design

The proposed system is a hierarchical cooperation orchestrated computing topology. We focus on improving the QoE by judiciously selecting the server actions. Different server and content selections affect the final server actions. Thus, to address the above complex optimization Eq. (13), we decompose it into two simpler subproblems as below.

1) Inner algorithm for recommendation list. First, we obtain the recommendation list K_m for the ID m from the content or data catalog, which is implemented by the collaborative filtering algorithm based on items-Inverse User Frequency (ItemCF-IUF). The inner algorithm is mainly divided into two steps: calculating the similarity between two contents and generating the recommendation list. When calculating the similarity, we consider the influence of the ID m activity on content similarity. We use the improved cosine formula to calculate the similarity between the content i and f as:

$$\mathrm{sim}_{i,f} = \frac{\sum_{m \in N_i \cap N_f} \dfrac{1}{\log^{1+|N(m)|}}}{\sqrt{|N_i| \cup |N_f|}}, \tag{14}$$

where N_i denotes the number of IDs that like the content i, N_f denotes the number of IDs that like the content f, and $|N_i| \cup |N_f|$ denotes the number of IDs that both like content i and f. Then the score of the content f will be calculated.

Then we sort $w_{m,f}$ in a descending order to generate the final recommendation list of the ID m. The details of the proposed method for solving the inner problem are shown in Algorithm 1. The internal of the loop consists of $|\mathcal{F}|$ calculations. Next, the complexity of the sorting step is $O(\log|\mathcal{F}|)$ in a pre-ordered list. Since these steps are repeated for every ID m, the total complexity of the algorithm is $O(|\mathcal{M}||\mathcal{F}|)$.

Algorithm 1: Inner Algorithm for Recommendation.

Input: $\mathcal{M}$, $\mathcal{F}$, and all IDs' information.
Output: $\{\boldsymbol{K}\}_{M \times R}$.
Initialization: $\{\boldsymbol{K}\}_{M \times R} \leftarrow 0_{M \times R}$;
for the ID $m \in \mathcal{M}$ **do**
  for each content pair of (i, f), $\forall i, f \in \mathcal{F}$ **do**
    Calculate $\mathrm{sim}_{i,f}$ and $w_{m,f}$;
    Sort $w_{m,f}$ in decreasing order;
    Choose the top R contents into the $\boldsymbol{K}_m$;
    Add $\boldsymbol{K}_m$ to $\{\boldsymbol{K}\}_{M \times R}$;
  end
end

2) Server actions. We optimize the server actions. As mentioned above, $\boldsymbol{\Pi}$ has $3MNF$ possible selections. It may be easy to find the optimal solution in a small scenario. Since the number of IDs, BSs, and contents can be large, it will take abundant time to converge if we use the general exhaustive methods (e.g., checking each combination of variables with a value of 0 or 1, and comparing the value of the objective function to obtain the optimal solution). To solve the problem, we propose a cache-friendly heuristic algorithm with the branch and bound (BNB) strategy.

Lemma 2: Eq. (13) can be divided into M independent subproblems as:

$$\mathcal{P}: \max Z_m = \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}} p_f\left(\mathrm{A}_{m,n,f}^1 \pi_{m,n,f}^L + \mathrm{A}_{m,n,f}^2 \pi_{m,n,f}^E + \mathrm{A}_{m,n,f}^3 \pi_{m,n,f}^C\right), \tag{15a}$$

s.t. the same with Problems (12b), (12c), (12d), (12e), and (12f). (15b)

Obviously, we have $Z^* = \sum_{m \in \mathcal{M}} Z_m$.

Proof: For each ID m, we seek the best strategy to satisfy its request and then it can benefit the whole cache system. Therefore, Eq. (13) can be separated, i.e., the sub-decision for each ID does not affect other IDs because there is no relevance between them.

Specifically, for a content or a data request generated by the ID m, we search server and content selections $\boldsymbol{\Pi}$ layer by layer. After initialization, we first determine whether a local direct hit occurs according to the cache state. If it does not happen, we consider whether the soft cache hits occur. If neither of the above two situations occurs, request processing will be completed through cooperative BSs or the CS. This procedure is repeated until the cache is full. To reduce unnecessary searches, we use the BNB strategy. In Eq. (15), when a feasible solution is determined by using the heuristic algorithm, the value of Z_m is calculated and denoted as Z_m^{ψ}. Thus, Z_m^{ψ} will be added to the constraint as the lower bound of the target value. Any solution with $Z_m < Z_m^{\psi}$ can be deleted without verifying whether it meets other constraints. By continuously improving the lower bound of the target value, the constraint conditions can be improved and the amount of calculation can be reduced.

The details of the proposed method for solving the whole problem are shown in Algorithm 2. And the computation complexity of Algorithm 2 is $O(|\mathcal{M}||\mathcal{N}||\mathcal{K}|)$.

Algorithm 2: Cache-Friendly Hierarchical Heuristic Algorithm.

Input: C, $\mathcal{N}$, $\mathcal{M}$, $\mathcal{F}$, $\mathcal{K}$, content request probability $\{p_f\}$, content cache status $\{s_{n,f}\}$.

Output: $\{\boldsymbol{\Pi}^*\}$.

Initialization: $\boldsymbol{\Pi}^L = 1$;
while the ID $m = 1, 2, ..., M$ **do**
  for the BS $n \in \mathcal{N}$ **do**
    for the content $f \in K_M$ **do**
      Calculate $Z_m(\pi_{m,n,f}^L)$ by Algorithm 1;
      Store it as Z_m^{ψ} in a sorted list;
      According to the cache state $s_{n,f}$, update $\pi_{m,n,f}$;
      Calculate $Z_m(\pi_{m,n,f})$;
      if $Z_m(\pi_{m,n,f}) > Z_m^{\psi}$ **then**
        Swap and update $\pi_{m,n,f}$;
        Add $\pi_{m,n,f}$ to $\boldsymbol{\Pi}$;
      end
  end
  if $\pi_{m,n,f} D_f > C$ then
    break;
  end
end
end
$\boldsymbol{\Pi} \leftarrow \text{argmax} Z^*(\boldsymbol{\Pi})$;
$\boldsymbol{\Pi}^* \leftarrow \boldsymbol{\Pi}$.

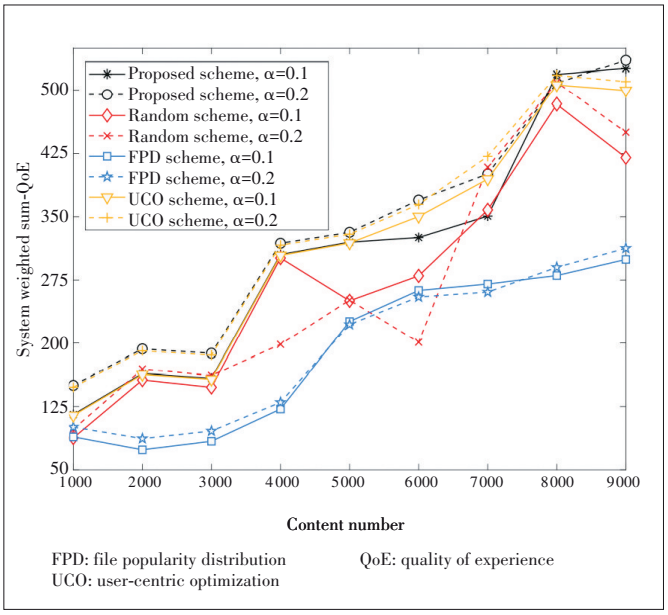
4 Simulation Results

For simulation purposes, all parameters are selected according to the real-world scenario. Numerical experiments are provided to evaluate the performance of the proposed scheme. We consider several BSs, each of which has the maximum coverage of a circle with a radius of 250 meters. And more than 400 IDs are randomly distributed within the coverage area of the BSs. We determine the local BS of each ID according to the association probability $a_{m,n}$. The channel gain is modeled as $g_{m,n} = 30.6 + 36.7 \log(l_{m,n})$ dB, where $l_{m,n}$ is the distance between the ID m and the BS n. The distance is randomly set as [0, 250] m. The wireless bandwidth, transmit power of each ID, and noise power is set as 20 MHz, [1.0, 1.5] W, and 10^{-13} W, respectively.

For IoT data, we consider a real data set consisting of 457 users and more than 9 000 video contents. And these contents are randomly cached in the BSs. The content size is randomly set as [2, 5] Mbit. Further, the cache constraint of the BS is set to a percentage θ of the total storage size. Besides, we use itemCF-IUF to get the recommendation list for each ID, and we get the corresponding score $w_{m,f}$. The parameter of Algorithm 1 is set as $R = 2$. To verify the experimental effect of the recommendation algorithm, we calculate the accuracy rate, recall rate, and their weighted harmonic average. And the results are respectively 0.4, 0.1311, and 0.1975.

To evaluate our proposed framework, we consider the following three baseline schemes: 1) File popularity distribution (FPD) strategy. As mentioned in Ref. [21], when a content request is generated by the ID, the cache system will distribute popular contents according to the popularity of contents. However, this strategy processes requests without considering content preferences and soft caching; 2) User-centric optimization (UCO) strategy. Similar to our paper, a simple QoE metric has been proposed for combining content caching with the recommender systems in Ref. [11]. They weigh the QoS and QoR, but the work of cooperative edge caching is missing; 3) Random scheme. The content request is randomly processed at the local BS, cooperative BSs, or the CS. $\boldsymbol{\Pi}$ is randomly set under the constraints in Eqs. (12d), (12e), and (12f).

In Fig. 2, we study different server selection schemes under contents ranging from 1 000 to 9 000, and eight independent simulations are considered (in this case, we set the $N = 2$). For each scheme, we set the balance constraint α to



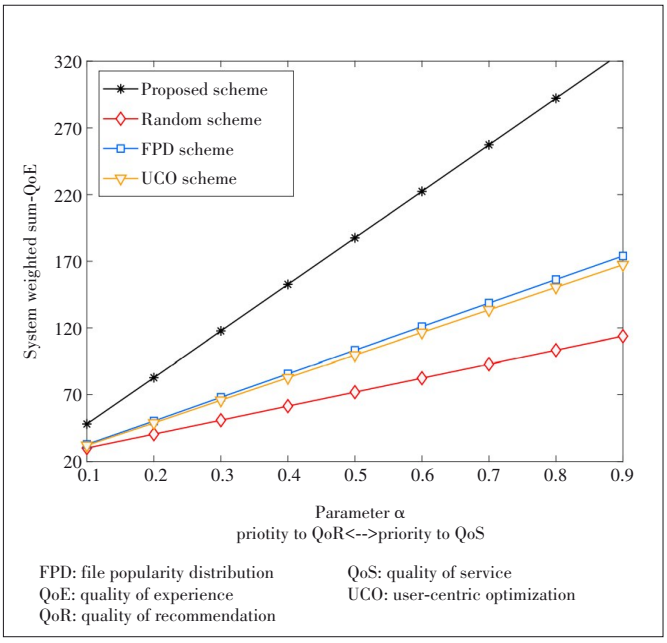
▲Figure 2. QoE versus different numbers of contents

0.1 and 0.2 respectively. We observe that the QoE increases rapidly with the increase of the contents in our proposed scheme, mainly because a tremendous amount of contents can provide more accurate references for recommendation (e.g., more historical behaviors). In the random scheme, the result fluctuates obviously because the decision is random. The experimental effect of our proposed scheme is also better than other schemes. In particular, the proposed scheme has an overall performance improvement of about 30% compared with the FPD scheme. The reason is that the soft cache fully considers content preferences, ensuring that content preferences are controllable and the distortion is minimized.

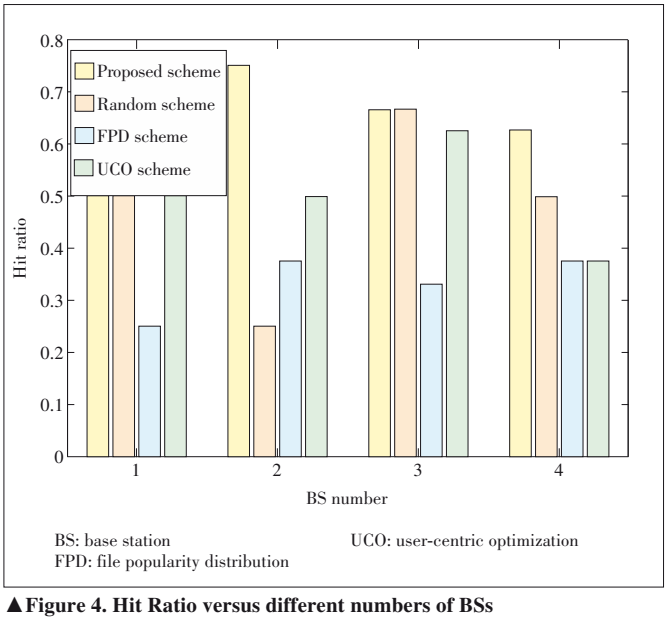
Next, we investigate whether the proposed scheme has better performance in QoS-QoR trade-off, as shown in Fig. 3. The balance factor α is in the range of 0.1 to 0.9. According to the simulation, the QoE increases linearly with the increase of α. When α = 0.1 (i.e., QoR is given priority), we observe that the performance of the FPD scheme and the UCO scheme is similar to the proposed scheme, mainly because cooperative caching has little effect on additional cache gain. When α increases gradually (i.e., a part of QoR is sacrificed and QoS is given priority), and the performance of the proposed scheme is greatly improved compared with the FPD scheme and UCO scheme. Due to the strong randomness of the random scheme, the performance improvement is not obvious.

We also evaluate the hit ratio under different BS numbers, as shown in Fig 4. In the proposed scheme, cache hits are defined as local hits and neighboring hits. We study different server selection schemes under the N range of 1 to 4. The hit ratio of the proposed scheme fluctuates depending on the number of BSs. For instance, it achieves the best hit ratio when the BS number is 2. But when the numbers of BSs are equal to 3 and 4, the hit ratio decreases gradually, mainly because more BSs will receive more content requests. In terms of improving the hit ratio, the performance of the proposed scheme is obviously better than the other three baseline schemes, mainly because the proposed scheme provides more cache hit possibilities.

The proposed scheme considers soft caching and the cooperation between the BSs. Compared with other baseline schemes, our proposed scheme considers the content prefer-



▲Figure 3. QoE versus different balance parameters



▲Figure 4. Hit Ratio versus different numbers of BSs

ences of the IDs to meet their needs and the BSs' cooperation to reduce the transmission delay of contents in the networks. Therefore, our scheme is superior to other schemes in the above comparative experiments.

5 Conclusions

In this paper, we have investigated the joint problem of cooperative edge caching and recommender systems for IoT systems. We have used the concept of soft caching by shifting from satisfying requests of IDs to satisfying their needs. Under the constraints of resources, computing conditions, etc., we choose the appropriate server actions to improve the QoE, which is defined as a 0 - 1 ILP problem. To solve it, we have proposed an uncomplicated and cache-friendly hierarchical heuristic algorithm with the BNB strategy. Simulation results have revealed the superior performance of the proposed scheme on increasing the QoE.

References

[1] MA H D, LIU L, ZHOU A F, et al. On networking of Internet of Things: explorations and challenges [J]. IEEE Internet of Things journal, 2016, 3(4): 441 - 452. DOI: 10.1109/JIOT.2015.2493082
[2] HE Y, YU F R, ZHAO N, et al. Software-defined networks with mobile edge computing and caching for smart cities: a big data deep reinforcement learning approach [J]. IEEE communications magazine, 2017, 55(12): 31 - 37. DOI: 10.1109/MCOM.2017.1700246
[3] GONG C, LIN F H, GONG X W, et al. Intelligent cooperative edge computing in Internet of Things [J]. IEEE Internet of Things journal, 2020, 7(10): 9372 - 9382. DOI: 10.1109/JIOT.2020.2986015
[4] CHEN B, LIU L, SUN M X, et al. IoT cache: toward data-driven network caching for Internet of Things [J]. IEEE Internet of Things journal, 2019, 6 (6): 10064 - 10076. DOI: 10.1109/JIOT.2019.2935442
[5] ZHANG F, HAN G J, LIU L, et al. Joint optimization of cooperative edge caching and radio resource allocation in 5G-enabled massive IoT networks [J]. IEEE Internet of Things journal, 2021, (99): 1. DOI: 10.1109/JIOT.2021.3068427
[6] LI X H, WANG X F, XIAO S J, et al. Delay performance analysis of cooperative cell caching in future mobile networks [C]//2015 IEEE International Conference on Communications. London, UK: IEEE, 2015: 5652 - 5657. DOI: 10.1109/ICC.2015.7249223
[7] LI X H, WANG X F, WAN P J, et al. Hierarchical edge caching in device-to-device aided mobile networks: modeling, optimization, and design [J]. IEEE journal on selected areas in communications, 2018, 36(8): 1768 - 1785. DOI: 10.1109/JSAC.2018.2844658
[8] YANG P, ZHANG N, ZHANG S, et al. Content popularity prediction towards location-aware mobile edge caching [J]. IEEE transactions on multimedia, 2019, 21(4): 915 - 929. DOI: 10.1109/TMM.2018.2870521
[9] SHANMUGAM K, GOLREZAEI N, DIMAKIS A G, et al. FemtoCaching: wireless content delivery through distributed caching helpers [J]. IEEE transactions on information theory, 2013, 59(12): 8402 - 8413. DOI: 10.1109/TIT.2013.2281606
[10] SERMPEZIS P, GIANNAKAS T, SPYROPOULOS T, et al. Soft cache hits: Improving performance through recommendation and delivery of related content [J]. IEEE journal on selected areas in communications, 2018, 36 (6): 1300 - 1313. DOI: 10.1109/JSAC.2018.2844983
[11] TSIGKARI D, SPYROPOULOS T. User-centric optimization of caching and recommendations in edge cache networks [C]//2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks". Cork, Ireland: IEEE, 2020: 244 - 253. DOI: 10.1109/WoWMoM49955.2020.00052
[12] CHATZIELEFTHERIOU L E, KARALIOPOULOS M, KOUTSOPOULOS I. Caching-aware recommendations: nudging user preferences towards better caching performance [C]//IEEE Conference on Computer Communications. Atlanta, USA: IEEE, 2017: 1 - 9. DOI: 10.1109/INFOCOM.2017.8057031
[13] SERMPEZIS P, SPYROPOULOS T, VIGNERI L, et al. Femto-caching with soft cache hits: Improving performance with related content recommendation [C]//IEEE Global Communications Conference. Singapore, Singapore: IEEE, 2017: 1 - 7. DOI: 10.1109/GLOCOM.2017.8254035
[14] LIU D, YANG C Y. A learning-based approach to joint content caching and recommendation at base stations [C]//IEEE Global Communications Conference. Abu Dhabi, United Arab Emirates: IEEE, 2018: 1 - 7. DOI: 10.1109/GLOCOM.2018.8647827
[15] GIANNAKAS T, SERMPEZIS P, SPYROPOULOS T. Show me the cache: optimizing cache-friendly recommendations for sequential content access [C]//2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks". Chania, Greece: IEEE, 2018: 14 - 22. DOI: 10.1109/WoWMoM.2018.8449731
[16] ZHENG D S, CHEN Y Y, YIN M X, et al. Cooperative cache-aware recommendation system for multiple Internet content providers [J]. IEEE wireless communications letters, 2020, 9(12): 2112 - 2115. DOI: 10.1109/LWC.2020.3014266
[17] COSTANTINI M, SPYROPOULOS T, GIANNAKAS T, et al. Approximation guarantees for the joint optimization of caching and recommendation [C]//IEEE International Conference on Communications. Dublin, Ireland: IEEE, 2020: 1 - 7. DOI: 10.1109/ICC40277.2020.9148740
[18] WANG F X, WANG F, LIU J C, et al. Intelligent video caching at network edge: a multi-agent deep reinforcement learning approach [C]//IEEE Conference on Computer Communications. Toronto, Canada: IEEE, 2020: 2499 - 2508. DOI: 10.1109/INFOCOM41043.2020.9155373
[19] SUN C, HUI L, LI X H, et al. Task offloading for end-edge-cloud orchestrated computing in mobile networks [C]//IEEE Wireless Communications and Networking Conference. Seoul, South Korea: IEEE, 2020: 1 - 6. DOI: 10.1109/WCNC45663.2020.9120496
[20] WANG X F, WANG C Y, LI X H, et al. Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching [J]. IEEE Internet of Things journal, 2020, 7(10): 9441 - 9455. DOI: 10.1109/JIOT.2020.2986803
[21] SUN R J, WANG Y, CHENG N, et al. QoE-driven transmission-aware cache placement and cooperative beamforming design in cloud-RANs [J]. IEEE transactions on vehicular technology, 2020, 69(1): 636 - 650. DOI: 10.1109/TVT.2019.2952726

Biographies

HAN Suning received the bachelor's degree in software engineering from Tiangong University, China in 2020. He is currently pursuing the master's degree with the School of Big Data and Software Engineering, Chongqing University, China. His current research interests include mobile edge computing, big data and recommender system.

LI Xiuhua (lixiuhua1988@gmail.com) received the B.S. degree from the Honors School, Harbin Institute of Technology, China in 2011, the M.S. de-

gree from the School of Electronics and Information Engineering, Harbin Institute of Technology, in 2013, and the Ph.D. degree from the Department of Electrical and Computer Engineering, The University of British Columbia, Canada in 2018. He joined Chongqing University through One-Hundred Talents Plan of Chongqing University, China in 2019. He is currently a tenure-track Assistant Professor with the School of Big Data & Software Engineering, and the Dean of the Institute of Intelligent Network and Edge Computing at Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University. His current research interests are 5G/B5G mobile Internet, mobile edge computing and caching, big data analytics, and machine learning.

SUN Chuan is a Ph.D. student with the School of Big Data & Software Engineering, Chongqing University, China. He received his B.S. degree from Wuhan University of Science and Technology, China in 2017. His current research interests include multi-access edge computing, recommender systems, and machine learning.

WANG Xiaofei is currently a professor with the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, China. He got his master's and doctoral degrees from Seoul National University, South Korea in 2006 and 2013, and was a postdoctoral fellow at The University of British Columbia, Canada from 2014 to 2016. Focusing on research of social-aware cloud computing, cooperative cell caching, and mobile traffic offloading, he has authored over 100 technical papers in *IEEE JSAC, IEEE TWC, IEEE Wireless Communications, IEEE Communications Magazine, IEEE TMM, IEEE INFOCOM, and IEEE SECON*. He was a recipient of the National Thousand Talents Plan (Youth) of China. He received the Scholarship for Excellent Foreign Students in the IT Field from NIPA of South Korea from 2008 to 2011, the Global Outstanding Chinese Ph.D. Student Award of the Ministry of Education of China in 2012, and the Peiyang Scholar of Tianjin University. In 2017, he received the Fred W. Ellersick Prize from the IEEE Communication Society.

Victor C. M. LEUNG is currently a Distinguished Professor of computer science and software engineering with Shenzhen University, China. He is also an Emeritus Professor of electrical and computer engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, The University of British Columbia, Canada. His research is in the broad areas of wireless networks and mobile systems. He has published widely in archival journals and refereed conference proceedings in these areas; several of his papers have won Best Paper Awards. He is a fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada.