IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 2, 2019, accepted August 16, 2019, date of publication August 22, 2019, date of current version September 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936866

Learning a Hybrid Proactive and Reactive Caching Policy in Wireless Edge Under Dynamic Popularity

KAIQIANG QI[1], SHENGQIAN HAN[1,2], AND CHENYANG YANG[1]
[1]School of Electronics and Information Engineering, Beihang University (BUAA), Beijing 100191, China
[2]Hangzhou Innovation Institute, Beihang University (BUAA), Hangzhou 310000, China

Corresponding author: Shengqian Han (sqhan@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61871015, and in part by the NSFC Key Project under Grant 61731002.

ABSTRACT Caching at wireless edge is a promising way to satisfy the explosively increasing mobile data demands, if future content popularity is known in advance. However, the time-varying nature of content popularity makes the popularity prediction far from perfect, which inevitably degrades the gain from caching. In this paper, we resort to a hybrid proactive and reactive policy to deal with the dynamics of popularity, in particular the hybrid of proactive probabilistic caching policy and least recently used policy, which are appropriate respectively for the contents with low and high dynamic popularity. We divide the contents requested in a region into two classes, where one can be modeled by independent reference model (IRM) and the other can be modeled by shot noise model (SNM). To maximize the total successful offloading ratio achieved by caching the two classes of contents, we optimize the hybrid caching policy including both the cache resource allocation to each class of contents and the probability of caching each IRM content. We find the optimal solution to the problem for general case, and provide a closed-form solution in a special case to gain insights. To provide a viable solution for practical use, we propose a heuristic method to obtain the optimal allocation fraction, and predict the popularity distribution and the allocation fraction using neural networks with historical data. We validate our analytical results by simulation results via synthetic datasets. We evaluate the performance of the proposed hybrid caching policy via synthetic data generated by SNM and two real datasets, and compare it with the proactive policy, the reactive policy and the existing hybrid proactive and reactive policy.

INDEX TERMS Dynamic popularity, proactive and reactive caching, cache resource allocation, shot noise model, independent reference model, real datasets.

I. INTRODUCTION

Caching at the wireless edge, e.g., base stations (BSs), has been shown a promising technique to support the explosively increasing traffic demand and improve user experience [1], [2], if future content popularity is known. Due to the vast number of available contents and the limited cache sizes at the BSs, proactive caching is believed of great potential for wireless edge caching [3]. As a consequence, proactive caching policies have been extensively studied towards diverse goals, such as minimizing average download delay [4], energy cost [5] or average service cost [6], and maximizing cache-hit probability [7], [8] or successful transmission probability [9].

Existing works on proactive caching have demonstrated remarkable performance gain. Most works, however, explicitly or implicitly assume that the requests for contents are subject to a so-called independent reference model (IRM) [10] or simply assume that the popularity is known. IRM assumes that the content catalogue is fixed, and the popularity distribution (i.e., the probability that a content is requested by all users in a region) is static. Although widely used and easy for optimization and analysis, IRM has been shown inappropriate to characterize the realistic arrival process of content requests recently [11], [12]. In practice, the popularity distribution is always time-varying, which is hard to predict accurately and

The associate editor coordinating the review of this article and approving it for publication was Walid Al-Hussaibi.

hence cannot be assumed known. This calls for machine learning techniques to make the prediction.

To deal with the dynamics of popularity distribution, several methods are proposed recently, which mainly include the following three kinds of approaches.

The first approach is to periodically predict the time-varying popularity distribution for updating proactive caching policies. In [13], a stacked auto-encoder based technique was proposed to classify the future popularity by extracting the spatio-temporal features of requests for contents. In [14], the number of requests in the next update duration was predicted with a grouped linear prediction method by taking the dynamic aging of each file into account. The caching performance of this kind of approach is limited by the inaccurate prediction of dynamic popularity, especially for the cold-start contents that will be requested in the next caching update duration but have never been requested before.

The second approach resorts to request-driven update of cache storage to better track the dynamics of popularity distribution. Different from traditional reactive caching polices, e.g., least recently used (LRU) or least frequently used policies, this approach employs reinforcement learning based methods, which is capable of learning the proactive caching decisions from past experiences. The works in [15], [16] directly optimized the caching policies with two kinds of deep reinforcement learning techniques, respectively. Nevertheless, this kind of approach is with high sample complexity and computational complexity.

The third approach is to design hybrid proactive and reactive caching, aimed to take the advantages of proactive caching meanwhile handle the contents whose request probabilities are hard or even impossible to predict. In [17], popularity distribution was predicted by extreme learning machine for periodically updating the cache list to initialize the cache space according to an optimized caching policy. A variation of LRU was then used to update the cache list in the next update duration. In [18], the cache resource was divided for proactive and reactive caching, where linear regression was used to predict the view counts of the music videos in the next day for proactive caching, and LRU was employed for reactive caching. The performance of this hybrid caching depends on how to allocate the cache resource, which however is not addressed in [18].

As observed in [19], a key feature of real-world content popularity is the so-called *temporal locality*, i.e., users' requests for a content almost all arrive within a short period, beyond which the content is rarely requested. The widely-employed IRM in the literature of wireless caching fails to model the temporal locality and hence is not applicable to characterize dynamic popularity. A novel traffic model, namely shot noise model (SNM), is more relevant, though more complicated, to characterize the users' requests [19]. SNM is able to capture several important content-level traffic features: the temporal locality of requests for a content, the time-varying content catalogue due to the birth of new contents and death of old contents and dynamic popularity distribution. In [20], SNM was employed to design the caching policy with small population of users at the wireless edge. In [21], SNM was used to generate synthetic data of requests for evaluating the caching performance.

Different from most priori works considering IRM (e.g., [4]–[9], [22]–[25]), which actually assumes that all contents have long lifespans and follow static popularity distribution with static catalogue size, we divide the contents requested in a region into two classes. To take into account the dynamic popularity distribution, we optimize a hybrid proactive and reactive caching policy for each BS with both IRM and SNM. To illustrate the gain of the hybrid caching policy in cache-enabled cellular networks, we employ the probabilistic caching policy commonly applied for randomly located BSs [9], [22] and LRU [19], [26] for the IRM and SNM contents, respectively, and use successful offloading ratio (SOR) as the objective for optimization. To remove the assumption of knowing future content popularity and other traffic parameters for optimizing the hybrid policy, we resort to neural networks to predict popularity distribution and the cache resource allocation with the historical data of requests.

The main contributions are summarized as follows.

- We formulate and solve an optimization problem for the hybrid caching policy, which jointly optimizes the fraction of cache resource allocated to each class of contents and the probability of caching each IRM content to maximize the total SOR. We derive a closed-form solution in a special case, which provides the guideline for designing an effective method to obtain the optimal cache resource allocation.
- We design neural networks to predict popularity distribution and learn the optimal cache resource allocation in the next update duration from the historical data.
- We evaluate the performance of the optimized hybrid caching policy with both synthetic and real datasets. Simulation results demonstrate evident gains of the proposed policy over non-hybrid proactive caching policy and reactive caching policy as well as the existing hybrid proactive and reactive caching policy on the datasets with dynamic popularity.

The remainder of this paper is organized as follows. Sec. II introduces the system and traffic models. Sec. III optimizes the hybrid caching policy, and provides a practical method to allocate cache resource. Sec. IV designs the learning-based hybrid caching policy. Sec. V evaluates the performance of the hybrid caching policy with both synthetic and real datasets. The conclusions are drawn in Sec. VI. The main parameters of the paper are summarized in TABLE 1.

II. SYSTEM AND TRAFFIC MODELS

Consider a cache-enabled wireless network, where the locations of BSs are modeled by a homogeneous Poisson point process (PPP), and users located according to an independent stationary point process initiate requests for contents. Each

TABLE 1. List of main parameters.

N_c	Cache resource of each BS
η	Fraction of cache resource allocated to SNM contents
c_f	Probability of caching the f-th IRM content
w^S	Fraction of average number of requests for SNM contents
q_f	Request probability of the f-th IRM content
N_f^S, N_f^I	Catalogue sizes of SNM and IRM contents
ρ_0	Average request arrival rate of IRM contents
$\Lambda_m(t)$	Popularity profile for the m-th SNM content at time instant t
V_m	Request volume for the m-th SNM content
λ	Average arrival rate of SNM contents
β	Pareto distribution parameter for SNM contents
δ^S, δ	Popularity skewness for SNM and IRM contents
T	Lifespan of each SNM content in the simplified SNM model
T_u	Caching update duration
T_o	Length of observation window

BS is connected to the core network with capacity-limited backhaul, and is equipped with cache that can store N_c contents. For simplicity, we assume that each content is with same size, but the results are applicable for the general case with different sizes by dividing each content into chunks with equal size.

A. TRAFFIC MODEL

The requested contents can be divided into two classes. One class of contents can be modeled by IRM (called IRM contents for short), and the other class of contents can be modeled by SNM (called SNM contents).

For the IRM class of contents, the requests for contents are assumed as a stationary Poisson request process with average request arrival rate ρ_0. The popularity distribution, defined as the request probabilities of contents in a catalogue of size N_f^I, follows Zipf distribution as in the literature [3], [27]. In particular, the probability of requesting the f-th most popular content is expressed as $q_f = f^{-\delta}/\sum_{j=1}^{N_f^I} j^{-\delta}$, where parameter δ reflects the popularity skewness and a larger value of δ indicates a more skewed popularity distribution. According to the analysis from real datasets, the value of δ is typically within (0, 1) [27].

For the SNM class of contents, the content-level traffic model proposed in [19] can be employed to characterize both the request arrival process of each content and the new content arrival process. In particular, the requests for the m-th content are modeled as an inhomogeneous Poisson process with the following features: 1) *popularity profile* $\Lambda_m(t)$, which is the normalized average request arrival rate at time instant t with $\Lambda_m(t) \geq 0$ and $\int_0^\infty \Lambda_m(\tau)\mathrm{d}\tau = 1$, 2) *lifespan* T_m, defined as $T_m = 1/\int_0^\infty \Lambda_m^2(\tau)\mathrm{d}\tau$ in [19], and 3) *volume of requests* V_m, which is the total number of requests received by the content during its lifespan. One can find that SNM models the time-varying average request arrival for each content, and then the instantaneous request arrival rate of the m-th content at t can be obtained as $V_m\Lambda_m(t - t_m)$, where t_m is the time when the content becomes available in the system and can be requested by users. The new content arrival is modeled as a homogeneous Poisson process with average arrival rate λ.

In [19], the request volume V_m is modeled as an independent identically distributed (i.i.d.) random variable following Pareto distribution with parameters $\beta > 1$ and $V_{\min}$, and the probability density function is

$$f_{V_m}(v) = \beta V_{\min}^{\beta} v^{-(\beta+1)}, \quad v \geq V_{\min}. \tag{1}$$

It is proved in [28] that if the values of V_m follow i.i.d. Pareto distribution, then the sorted volumes of requests in descending order approximately obey the Zipf distribution with parameter $\delta^S = \beta^{-1}$ for a large content catalogue. Therefore, we can use parameter δ^S to reflect the popularity skewness of the SNM contents.

For the SNM contents, the analyses in [19], [29] indicate that the performance of LRU policy mainly depends on the average lifespan of the contents and is not sensitive to the shape of popularity profile $\Lambda_m(t)$. This suggests that we can employ a simplified SNM proposed in [20]. It assumes that all the SNM contents have equal lifespan T, which is the average lifespan of all SNM contents, and assumes a rectangular popularity profile for all contents with $\Lambda(t) = 1/T$, $t \in [0, T]$. With this simplified SNM, we can obtain the average number of requested contents during the lifespan as $N_f^S = \lambda T$, which can be regarded as the catalogue size of the SNM contents.

An example of the SNM is provided in Fig. 1, where the rectangular popularity profile is used for illustration.

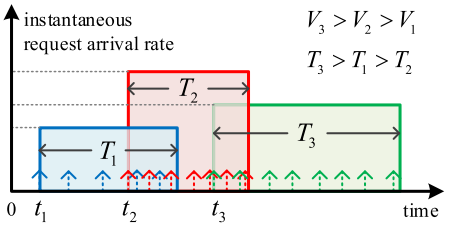


FIGURE 1. Illustration of the arrival process of three SNM contents and their request arrival processes with rectangular popularity profile.

Remark 1: The assumptions in this subsection are only used for deriving and analyzing the optimal hybrid caching policy, which are no longer used again later when we learn the popularity distribution and the cache resource allocation from the historical data.

B. USER ASSOCIATION AND PERFORMANCE METRIC

We consider densely deployed BSs, where the system is interference-limited and the impact of noise is negligible. For cache-enabled cellular networks where only one class of contents are requested, successful offloading probability (SOP) is often employed as the performance metric for edge caching. The SOP is defined as the probability that the requested content from a user can be successfully downloaded from the cache of a BS. For interference-limited networks, successfully downloading means that the received signal-to-interference ratio (SIR) is larger than a given threshold γ_0.

The SOP depends on *user association strategy*. Under the hybrid caching policy, each BS maintains two lists of cached contents, namely IRM list and SNM list. The IRM list is generated by the proactive probabilistic caching policy, which can be updated in off-peak time, say midnight. Thus, it is reasonable to assume that each BS can share the IRM list with other BSs. The SNM list is generated by the reactive LRU policy, which is frequently updated by evicting the least recently used contents whenever new requests arrive. Due to the difficulty of sharing the dynamic SNM list among BSs, we assume that each BS does not know the SNM lists of other BSs. This leads to the following user association strategy. If a user requests an IRM content, its local BS (i.e., its nearest BS) will check the IRM lists of all BSs. If the requested content is on the lists meanwhile satisfying the SIR threshold, the user will be associated with the nearest BS that caches the requested content. Otherwise, the user will be associated with its local BS to fetch the content via backhaul. If a user requests a SNM content, it will be associated with its local BS since the SNM lists of other BSs are unknown. The local BS fetches the requested content either from its own cache or via backhaul, depending on whether or not the requested content is on its SNM list.

Since we consider cellular networks where two classes of contents are requested, we employ total SOR achieved by caching both SNM and IRM contents as the objective function. The SOR is defined as the ratio of the average numbers of successfully offloaded requests for both classes of contents to the average number of all received requests in a caching update duration, which is a weighted sum of the SOP achieved by each class of contents.

III. OPTIMIZING A HYBRID PROACTIVE AND REACTIVE CACHING POLICY

The caching policies appropriate for the IRM and SNM contents differ. For the IRM contents, the static popularity distribution can be predicted from the historical number of requests, and hence proactive caching policy can perform well. For the SNM contents, the popularity distribution is hard to predict, especially for the newly-arrived contents (i.e., cold-start contents). This inevitably degrades the performance of proactive caching, and hence reactive caching policy is often employed [19]. Since both IRM and SNM contents may be requested in a cell, it is natural to design a hybrid policy with both proactive and reactive caching, where the cache resource at a BS is periodically allocated to cache IRM and SNM contents. To illustrate the benefits of the hybrid proactive and reactive caching, we take two typical caching policies as an example, which are the proactive probabilistic caching policy appropriate for PPP-modeled BSs [9], [22] and the reactive LRU caching policy.

In this section, we formulate and solve an optimization problem for the hybrid caching policy, which determines the cache resource allocation to the two classes of contents and the probability of caching each IRM content. We then derive a closed-form solution in a special case.

A. PROBLEM FORMULATION

Let $p_{\text{off}}^{\text{S}}(\eta)$ denote the SOP achieved by caching the SNM contents, which depends on the fraction of cache resource η allocated to this class of contents, $0 \le \eta \le 1$. Then, the fraction of cache resource allocated to the IRM contents is $1-\eta$. Let $p_{\text{off}}^{\text{I}}(\eta, \mathbf{c})$ denote the SOP of caching the IRM contents with the proactive probabilistic caching policy [9], [22], which depends on η and the probabilities of caching the IRM contents, denoted as $\mathbf{c} = [c_f]_{f=1,\cdots,N_{\text{f}}^{\text{I}}}$ with $0 \le c_f \le 1$ and $\|\mathbf{c}\|_1 \le (1-\eta)N_{\text{c}}$, where $\|\cdot\|_1$ is the l_1 norm.

The total SOR achieved by caching both SNM and IRM contents can be obtained as

$$\begin{aligned} p_{\text{off}}^{\text{tot}}(\eta, \mathbf{c}) &= \frac{\mathbb{E}[Q^{\text{S}}]p_{\text{off}}^{\text{S}}(\eta) + \mathbb{E}[Q^{\text{I}}]p_{\text{off}}^{\text{I}}(\eta, \mathbf{c})}{\mathbb{E}\left[Q^{\text{S}} + Q^{\text{I}}\right]} \\ &\triangleq w^{\text{S}} p_{\text{off}}^{\text{S}}(\eta) + w^{\text{I}} p_{\text{off}}^{\text{I}}(\eta, \mathbf{c}), \end{aligned} \tag{2}$$

where Q^{S} and Q^{I} respectively denote the numbers of received requests for the SNM and IRM contents during each caching update duration T_{u}, $w^{\text{S}} = \frac{\mathbb{E}[Q^{\text{S}}]}{\mathbb{E}[Q^{\text{S}}+Q^{\text{I}}]}$ and $w^{\text{I}} = \frac{\mathbb{E}[Q^{\text{I}}]}{\mathbb{E}[Q^{\text{S}}+Q^{\text{I}}]}$ are respectively the fractions of the average numbers of requests for the SNM and IRM contents during T_{u}, and $\mathbb{E}[\cdot]$ denotes the expectation operation. It is clear that $w^{\text{S}} + w^{\text{I}} = 1$.

For the SNM contents, the number of new contents in T_{u} (i.e., the contents that have never been requested before), denoted by n, is a random variable, which obeys Poisson distribution with mean λT_{u}. Since V_m, $m = 1, \cdots, n$ are i.i.d. random variables, the average number of requests for the SNM contents can be expressed by taking the expectation over both n and V_m as

$$\begin{aligned} \mathbb{E}\left[Q^{\text{S}}\right] &= \mathbb{E}_{V_m}\left\{\mathbb{E}_n\left[\sum_{m=1}^{n} V_m\right]\right\} \\ &= \mathbb{E}_n\left\{n\mathbb{E}[V_m]\right\} = \lambda T_{\text{u}}\mathbb{E}[V_m]. \end{aligned} \tag{3}$$

For the IRM contents, the average number of requests during T_{u} is $\mathbb{E}[Q^{\text{I}}] = \rho_0 T_{\text{u}}$. Then, we have

$$w^{\text{S}} = \frac{\lambda\mathbb{E}[V_m]}{\lambda\mathbb{E}[V_m] + \rho_0}, \quad w^{\text{I}} = \frac{\rho_0}{\lambda\mathbb{E}[V_m] + \rho_0}. \tag{4}$$

To obtain the expression of $p_{\text{off}}^{\text{tot}}(\eta, \mathbf{c})$ for optimization, we next investigate the SOPs achieved by caching IRM and SNM contents separately.

Given the considered user association strategy, the SOP achieved by caching the IRM contents with probabilistic caching policy in systems subject to Rayleigh fading channel can be obtained as

$$p_{\text{off}}^{\text{I}}(\eta, \mathbf{c}) = \sum_{f=1}^{N_{\text{f}}^{\text{I}}} q_f \mathbb{P}\{\gamma_f > \gamma_0\} \overset{(a)}{=} \sum_{f=1}^{N_{\text{f}}^{\text{I}}} \frac{q_f c_f}{(1-\gamma_0^{2/\alpha}\epsilon_1)c_f + \gamma_0^{2/\alpha}\epsilon_0}, \tag{5}$$

where γ_f is the SIR at the user when receiving the f-th IRM content, $\mathbb{P}\{\gamma_f > \gamma_0\}$ is the probability of γ_f being larger than the SIR threshold γ_0, step (a) is derived using the similar method in [24], α is the path-loss exponent,

$\epsilon_0 = \int_0^{\infty} \frac{1}{1+u^{\alpha/2}} \mathrm{d}u$, $\epsilon_1 = \int_0^{\gamma_0^{-2/\alpha}} \frac{1}{1+u^{\alpha/2}} \mathrm{d}u$, and the term $1-\gamma_0^{2/\alpha}\epsilon_1 > 0$ holds because $\gamma_0^{2/\alpha}\epsilon_1 = \int_0^1 \frac{\gamma_0}{\gamma_0+v^{\alpha/2}} \mathrm{d}v < \int_0^1 \mathrm{d}v = 1$.

The SOP for the SNM contents can be expressed as

$$p_{\mathrm{off}}^{\mathrm{S}}(\eta) = p_{\mathrm{h}}^{\mathrm{S}}(\eta)\mathbb{P}\{\gamma > \gamma_0\} \overset{(b)}{=} \frac{p_{\mathrm{h}}^{\mathrm{S}}(\eta)}{1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1)}, \quad (6)$$

where $p_{\mathrm{h}}^{\mathrm{S}}(\eta)$ is the cache-hit ratio of LRU policy, $\mathbb{P}\{\gamma > \gamma_0\}$ is the probability of the SIR $\gamma > \gamma_0$, and again step (b) can be obtained by following the derivations in [24] given the user association strategy.

The cache-hit ratio of LRU policy depends on cache eviction time, which is the duration from the time when a content enters the cache until the content is evicted. The cache eviction time is a random variable relying on the request arrival process of the content. To derive the cache-hit ratio of LRU, Che's approximation is commonly applied by assuming that the cache eviction time is constant and independent of specific contents [10]. Based on this approximation, the cache-hit ratio for the SNM contents with a single cache can be approximated as [19]

$$p_{\mathrm{h}}^{\mathrm{S}}(\eta) \approx 1 - \int_0^{\infty} \Lambda(\tau) \frac{\phi'_{V_m}\left(-\int_0^{t_{\mathrm{c}}} \Lambda(\tau-\theta)\mathrm{d}\theta\right)}{\mathbb{E}[V_m]} \mathrm{d}\tau, \quad (7)$$

where $\phi'_{V_m}(x) = \mathbb{E}_{V_m}\left[V_m e^{xV_m}\right]$, and t_{c} is the cache eviction time for the SNM contents, which is the unique solution of the following equation

$$\eta N_{\mathrm{c}} = \lambda \int_0^{\infty} 1 - \phi_{V_m}\Big(-\int_0^{t_{\mathrm{c}}} \Lambda(\tau-\theta)\mathrm{d}\theta\Big)\mathrm{d}\tau, \quad (8)$$

where $\phi_{V_m}(x) = \mathbb{E}_{V_m}\left[e^{xV_m}\right]$.

The results given by (7) and (8) cannot be directly used for the optimization of the hybrid caching policy, because they do not provide an explicit relationship between $p_{\mathrm{h}}^{\mathrm{S}}(\eta)$ and η. In the following, we develop a closed-form expression of $p_{\mathrm{h}}^{\mathrm{S}}(\eta)$ with respect to η by resorting to approximations under the small-cache scenario, where the cache size of a BS is far smaller than the content catalogue size, i.e., $N_{\mathrm{c}} \ll \min\left\{N_{\mathrm{f}}^{\mathrm{S}}, N_{\mathrm{f}}^{\mathrm{I}}\right\}$. The approximated cache-hit ratio is given in the following proposition.

Proposition 1: The cache-hit ratio for the SNM contents under the small-cache scenario can be approximated as

$$p_{\mathrm{h}}^{\mathrm{S}}(\eta) \approx 1 - \frac{\mathbb{E}\left[V_m \int_0^{\infty} \Lambda(\tau) e^{-\frac{V_m \Lambda(\tau)}{\lambda \mathbb{E}[V_m]}\eta N_{\mathrm{c}}} \mathrm{d}\tau\right]}{\mathbb{E}[V_m]}. \quad (9)$$

Proof: See Appendix A. ∎

Note that *Proposition 1* is also applicable to our network with multiple caches where the traffic load is uniform among cells.

By substituting (9) into (6), we can obtain the SOP of the SNM contents $p_{\mathrm{off}}^{\mathrm{S}}(\eta)$. Then, substituting it and (5) into (2), we can approximate the total SOR as

$$p_{\mathrm{off}}^{\mathrm{tot}}(\eta, \mathbf{c}) \approx \frac{w^{\mathrm{S}}}{1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1)}\left(1 - \frac{\mathbb{E}\left[V_m e^{-\frac{V_m}{N_{\mathrm{f}}^{\mathrm{S}}\mathbb{E}[V_m]}\eta N_{\mathrm{c}}}\right]}{\mathbb{E}[V_m]}\right) + w^{\mathrm{I}} \sum_{f=1}^{N_{\mathrm{f}}^{\mathrm{I}}} q_f \frac{c_f}{c_f + \gamma_0^{2/\alpha}(\epsilon_0 - c_f\epsilon_1)} \triangleq \hat{p}_{\mathrm{off}}^{\mathrm{tot}}(\eta, \mathbf{c}), \quad (10)$$

where the rectangular popularity profile $\Lambda(\tau) = 1/T$ for $\tau \in [0, T]$ is considered as discussed in Sec. II.

Remark 2: The SOP for SNM contents given in (6) (and (9)) depends on several traffic parameters, e.g., the average content arrival rate λ, average lifespan T and Pareto distribution parameter β. This seems to conflict with the usage of LRU for SNM contents since LRU policy does not need any traffic parameters. It should be clarified that these parameters are only used for optimizing η rather than the caching policy.

Finally, we can formulate the hybrid caching policy optimization problem as follows,

$$\begin{aligned} \mathsf{P1}: \ & \max_{\eta, \mathbf{c}} \ \hat{p}_{\mathrm{off}}^{\mathrm{tot}}(\eta, \mathbf{c}) && (11a)\\ & s.t. \ 0 \le \eta \le 1, && (11b)\\ & \quad 0 \le c_f \le 1, \quad f = 1, \cdots, N_{\mathrm{f}}^{\mathrm{I}}, && (11c)\\ & \quad \eta N_{\mathrm{c}} + \|\mathbf{c}\|_1 \le N_{\mathrm{c}}, && (11d) \end{aligned}$$

where (11d) is the constraint on cache resource, the term ηN_{c} is the cache resource for SNM contents, and the term $\|\mathbf{c}\|_1$ is the cache resource for IRM contents as proved in [22].

In practice, we need to ensure that the cache resource allocated to each class of contents is an integer, i.e., $\eta N_{\mathrm{c}} \in \mathbb{Z}$ and $\|\mathbf{c}\|_1 \in \mathbb{Z}$. Under the integer constraint, we can find the optimal values of η and $\mathbf{c}$ that maximize $\hat{p}_{\mathrm{off}}^{\mathrm{tot}}(\eta, \mathbf{c})$ in two steps: 1) obtain the optimal solution η^* and $\mathbf{c}^*$ of $\mathsf{P1}$, 2) quantize $\eta^* N_{\mathrm{c}}$ and $\|\mathbf{c}^*\|_1$ to address the integer constraint by providing a quantization algorithm. The details are presented in the next subsection.

B. OPTIMIZATION OF HYBRID CACHING POLICY

Since the term $1-\gamma_0^{2/\alpha}\epsilon_1$ is positive as shown below (5), we can readily prove that the objective function of problem $\mathsf{P1}$ is concave by examining its second derivative. From the Karush-Kuhn-Tucker (KKT) conditions of $\mathsf{P1}$ [30], the optimal solution η^* and $\mathbf{c}^*$ of $\mathsf{P1}$ can be derived as (see Appendix B)

$$\eta^* = \left[g_{\mathrm{s}}^{-1}\left(\mu^*\left(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1)\right)\right)\right]_0^1, \quad (12a)$$

$$c_f^* = \left[\frac{1}{1-\gamma_0^{2/\alpha}\epsilon_1}\left(\left(\frac{\gamma_0^{2/\alpha}\epsilon_0 w^{\mathrm{I}} q_f}{\mu^*}\right)^{1/2} - \gamma_0^{2/\alpha}\epsilon_0\right)\right]_0^1, \quad \forall f, \quad (12b)$$

where $g_{\mathrm{s}}(\eta^*) \triangleq \frac{w^{\mathrm{S}}}{N_{\mathrm{f}}^{\mathrm{S}}\mathbb{E}^2[V_m]}\mathbb{E}\{V_m^2 \, e^{-\frac{V_m \eta^* N_{\mathrm{c}}}{N_{\mathrm{f}}^{\mathrm{S}}\mathbb{E}[V_m]}}\}$ monotonically decreases with η^*, $g_{\mathrm{s}}^{-1}(\cdot)$ is the inverse function of $g_{\mathrm{s}}(\cdot)$, $[x]_0^1 = \min\{\max\{x, 0\}, 1\}$ denotes that x is truncated by 0 and 1, and μ^* can be obtained from the equality $\eta^* N_{\mathrm{c}} + \|\mathbf{c}^*\|_1 = N_{\mathrm{c}}$ by bisection search. It is noteworthy that (12b) is applicable to the case with any given allocation fraction.

After obtaining the optimal solution of **P1**, η^* and $\mathbf{c}^*$, we next quantize $\eta^* N_\mathrm{c}$ and $\|\mathbf{c}^*\|_1$ to address the integer constraint. We first compute the SORs corresponding to the ceiling and floor of $\eta^* N_\mathrm{c}$ as $\hat{p}_\mathrm{off}^\mathrm{tot}(\eta_1, \mathbf{c}_1)$ and $\hat{p}_\mathrm{off}^\mathrm{tot}(\eta_2, \mathbf{c}_2)$, respectively, where $\eta_1 = \lceil \eta^* N_\mathrm{c} \rceil / N_\mathrm{c}$, $\eta_2 = \lfloor \eta^* N_\mathrm{c} \rfloor / N_\mathrm{c}$, $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are the ceiling and floor operators, and $\mathbf{c}_1$ and $\mathbf{c}_2$ that satisfy $\|\mathbf{c}_1\|_1 = (1-\eta_1)N_\mathrm{c}$ and $\|\mathbf{c}_2\|_1 = (1-\eta_2)N_\mathrm{c}$ can be found by bisection search, respectively. Then, we compare $\hat{p}_\mathrm{off}^\mathrm{tot}(\eta_1, \mathbf{c}_1)$ and $\hat{p}_\mathrm{off}^\mathrm{tot}(\eta_2, \mathbf{c}_2)$, and choose the one (i.e., η_1 and $\mathbf{c}_1$, or η_2 and $\mathbf{c}_2$) with larger SOR as the final solution.

C. SPECIAL-CASE ANALYSIS

To gain useful insights, we derive a closed-form solution of cache resource allocation in a special case, where the SIR threshold γ_0 is relatively high, say no less than 0 dB, such that each user needs to be associated with the local BS with high probability. Given the local BS association strategy, it is shown in [1] that the optimal proactive caching policy is the *popular caching policy*, i.e., deterministically caching the most popular contents. This means that in this special case the probabilistic caching policy becomes the popular caching policy for the IRM contents.

With such caching policy, only the $(1-\eta)N_\mathrm{c}$ most popular contents are cached, i.e., the caching probabilities are $c_f = 1$ for $f = 1, \cdots, (1-\eta)N_\mathrm{c}$ and $c_f = 0$ for other f. Then, the SOP of the IRM contents can be obtained as

$$p_\mathrm{off}^\mathrm{I}(\eta) = \frac{\sum_{f=1}^{(1-\eta)N_\mathrm{c}} q_f}{1 + \gamma_0^{2/\alpha}(\epsilon_0 - \epsilon_1)} = \frac{p_\mathrm{h}^\mathrm{I}(\eta)}{1 + \gamma_0^{2/\alpha}(\epsilon_0 - \epsilon_1)}, \quad (13)$$

where $p_\mathrm{h}^\mathrm{I}(\eta) \triangleq \sum_{f=1}^{(1-\eta)N_\mathrm{c}} q_f$ is the cache-hit ratio.

For notational simplicity, suppose that the SNM contents and the IRM contents exhibit the same popularity skewness, i.e., $\delta^\mathrm{S} = \delta$, but the analysis for arbitrary δ^S and δ is similar.

By substituting (13) into **P1**, we can obtain a problem to optimize the allocation of cache resource η. Yet because η locates in the upper limit of the summation in the numerator of (13), solving the problem is difficult. We thus introduce the approximation $\sum_{i=1}^{M} i^{-\delta} \approx \int_0^M x^{-\delta} \mathrm{d}x = \frac{M^{1-\delta}}{1-\delta}$ for $\delta \in (0, 1)$, which is an inverse operation of numerical integration. Then, $p_\mathrm{h}^\mathrm{I}(\eta)$ can be approximated as

$$p_\mathrm{h}^\mathrm{I}(\eta) = \frac{\sum_{f=1}^{(1-\eta)N_\mathrm{c}} f^{-\delta}}{\sum_{j=1}^{N_\mathrm{f}^\mathrm{I}} j^{-\delta}} \approx \left(\frac{(1-\eta)N_\mathrm{c}}{N_\mathrm{f}^\mathrm{I}}\right)^{1-\delta} \triangleq \hat{p}_\mathrm{h}^\mathrm{I}(\eta), \quad (14)$$

which is accurate for small or moderate popularity skewness δ according to our evaluations.

With (13) and (14), **P1** can be rewritten as

$$\mathbf{P2}: \max_{\eta} p_\mathrm{off}^\mathrm{tot}(\eta) \approx \frac{w^\mathrm{S}\hat{p}_\mathrm{h}^\mathrm{S}(\eta) + w^\mathrm{I}\hat{p}_\mathrm{h}^\mathrm{I}(\eta)}{1 + \gamma_0^{2/\alpha}(\epsilon_0 - \epsilon_1)} \quad (15a)$$

$$s.t.\ 0 \le \eta \le 1. \quad (15b)$$

It is not hard to prove that **P2** is concave. Then, from the KKT conditions, we can obtain the first-order optimality condition of **P2** as

$$-\frac{w^\mathrm{S}N_\mathrm{c}}{N_\mathrm{f}^\mathrm{S}\mathbb{E}^2[V_m]}\mathbb{E}\left[V_m^2\, e^{-\frac{V_m\eta^*N_\mathrm{c}}{N_\mathrm{f}^\mathrm{S}\mathbb{E}[V_m]}}\right] + w^\mathrm{I}(1-\delta)\left(\frac{N_\mathrm{c}}{N_\mathrm{f}^\mathrm{I}}\right)^{1-\delta}\left(1-\eta^*\right)^{-\delta} \begin{cases} = 0, & \text{if } 0 < \eta^* < 1, \\ \ge 0, & \text{if } \eta^* = 0, \\ \le 0, & \text{if } \eta^* = 1, \end{cases} \quad (16)$$

where the third case will not happen for $0 < \delta < 1$, because with $\eta^* = 1$ the left-hand side of (16) is positive. It indicates that only caching SNM contents at BS is not optimal.

Upon substituting $\delta = \delta^\mathrm{S} = \beta^{-1}$ and (4) into (16), we obtain that

$$\underbrace{\frac{\beta/(1-\beta)}{\mathbb{E}[V_m]T}\mathbb{E}\left[V_m^2 e^{-\frac{V_m}{N_\mathrm{f}^\mathrm{S}\mathbb{E}[V_m]}\eta^*N_\mathrm{c}}\right]}_{g_l(\eta^*)} - \underbrace{\frac{\rho_0}{N_\mathrm{f}^\mathrm{I}}\left(\frac{(1-\eta^*)N_\mathrm{c}}{N_\mathrm{f}^\mathrm{I}}\right)^{-\frac{1}{\beta}}}_{g_r(\eta^*)} \begin{cases} = 0, & \text{if } 0 < \eta^* < 1, \\ \le 0, & \text{if } \eta^* = 0, \end{cases} \quad (17)$$

where two functions $g_l(\eta^*)$ and $g_r(\eta^*)$ are defined as shown in (17). Then, the optimal solution η^* can be approximated as shown in the following proposition.

Proposition 2: The optimal solution η^* under the small-cache scenario can be approximated as

$$\eta^* \approx \begin{cases} h^{-1}(DN_\mathrm{c}), & \text{if } 1 < \beta < 2, \\ \max\left\{1 - \left(\frac{\rho_0 T}{\mathbb{E}[V_m]}\frac{\beta-2}{\beta-1}\right)^{\beta}\frac{(N_\mathrm{f}^\mathrm{I})^{1-\beta}}{N_\mathrm{c}}, 0\right\}, & \text{if } \beta > 2, \end{cases} \quad (18)$$

where $D \triangleq \left(\left(\frac{A_\beta \mathbb{E}[V_m]}{\rho_0 T}\right)^{\beta}(N_\mathrm{f}^\mathrm{I})^{\beta-1}(\lambda T)^{(2-\beta)\beta}\right)^{\frac{1}{(\beta-1)^2}} > 0$ for $1 < \beta < 2$ with $A_\beta = \beta\left(\frac{\beta-1}{\beta}\right)^{\beta-1}\Gamma(2-\beta)$, $h(x) = x^{\frac{(2-\beta)\beta}{(\beta-1)^2}}/(1-x)^{\frac{1}{(\beta-1)^2}}$, $h^{-1}(\cdot)$ is the inverse function of $h(\cdot)$, and $h^{-1}(DN_\mathrm{c})$ monotonically increases with N_c.

Proof: See Appendix C. ■

Remark 3: Based on the closed-form solution (18), we can observe several monotonic properties of the optimal fraction of cache resource allocated to SNM contents η^* as follows. It can be observed that η^* increases with the average request volume of the SNM content $\mathbb{E}[V_m]$, and increases with the average content arrival rate of the SNM contents λ if $1 < \beta < 2$. In addition, η^* decreases with the average request arrival rate of the IRM contents ρ_0. From (4), we can see that w^S also increases with $\mathbb{E}[V_m]$ and λ, and decreases with ρ_0. This observation is helpful to design a practical cache allocation fraction method with learned information from historical number of requests.

Besides, from *Proposition 2* we can also obtain the following observations that help understand the behavior of the optimal cache resource allocation. (1) η^* increases with the cache size of BS N_c if $\eta^* \neq 0$. This can be explained as

follows. With the popular caching policy, the IRM contents with high request probabilities will be cached first. When further increasing the cache size, since the remaining IRM contents are less popular, the fraction of cache resource allocated to them should be less than that allocated to the already cached popular contents. As a result, the total fraction of cache resource allocated to the IRM contents will reduce when N_c increases, i.e., η^* will increase. (2) η^* increases with the catalogue size of the IRM contents N_f^I. This is because once ρ_0 is given, the average request arrival rate of each IRM content decreases with N_f^I, leading to the decreasing fraction of cache resource allocated to the IRM contents.

D. NUMERICAL AND SIMULATION RESULTS WITH SYNTHETIC DATASET

In this subsection, we first evaluate the accuracy of the introduced approximations. With the observations in *Remark 3*, we propose a heuristic but effective method to obtain the optimal allocation fraction.

To this end, we employ a controllable dataset, which is synthesized according to the traffic model presented in Sec. II with the parameters listed in TABLE 2. The parameters are used throughout this subsection if not otherwise specified. The simulation results are averaged over 100 Monte-Carlo trails. In each trial, we synthesize the content requests independently, and the SIR distribution for each request is computed by $\mathbb{P}\{\gamma_f > \gamma_0\}$ in step (a) of (5) for the IRM content, or $\mathbb{P}\{\gamma > \gamma_0\}$ in step (b) of (6) for the SNM content. It is shown that both (5) and (6) are irrelevant to the density and transmit power of the BSs for the considered interference-limited network.

TABLE 2. Simulation parameters.

Path-loss exponent, α	3.7
Received SIR threshold at users, γ_0	-10 dB [31]
Cache size of BS, N_c	100 contents
Average request volume of each SNM content during lifespan, $\mathbb{E}[V_m]$	3 requests [19]
Average request arrival rate of IRM contents, ρ_0	1.5×10^3 requests/day
Catalogue size of IRM contents, N_f^I	5×10^3 contents
Popularity skewness, $\delta = \beta^{-1}$	$\frac{2}{3}$
Average content arrival rate, λ	10^3 contents/day
Average lifespan for SNM contents, T	5 days
Caching update duration, T_u	1 day

1) ACCURACY OF THE APPROXIMATIONS

In Fig. 2, we evaluate the accuracy of the approximations used to derive the objective function of problem **P1**. The legend "Num" denotes the numerical solutions obtained from (12). The legend "Simu" denotes the solution obtained by simulation without any approximations, where we employ exhaustive search with step size 0.02 to find the value of η achieving the maximal total SOR. As shown in Fig. 2(a), the values of η^* obtained by numerical solutions and simulations are close especially for small cache size N_c. As shown

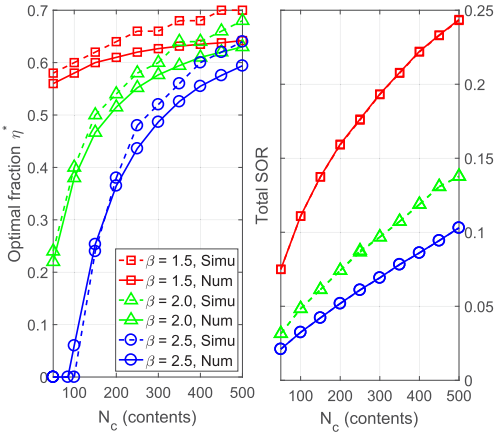


FIGURE 2. Accuracy of the approximations used in problem P1. (a) Optimal fraction of cache resource allocation. (b) Total SOR achieved by η^*.

in Fig. 2(b), the simulation and numerical results of the total SORs achieved by η^* overlap even for large cache size despite that in this case the numerically obtained value of η^* has a gap from the simulation result without any approximation. This indicates that the employed approximations, including Che's approximation and the approximation applied in *Proposition 1*, have negligible impact on the performance of the hybrid caching policy.

We next evaluate the accuracy of the approximations used to obtain the closed-form solution in (18) under the special case in Sec. III-C.

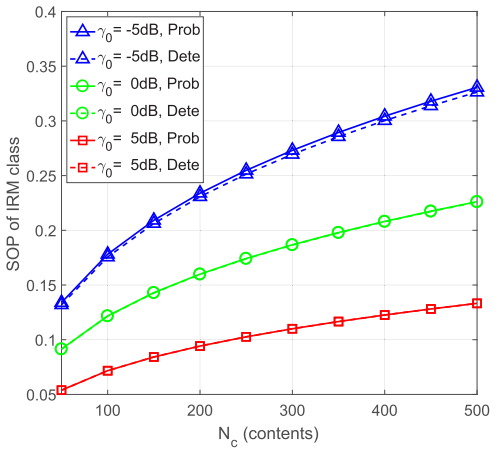


FIGURE 3. Probabilistic v.s. deterministic caching policy.

In Fig. 3, we provide the SOPs of the IRM contents achieved by using the deterministic popular caching policy and the probabilistic caching policy, which are denoted by the legends "Dete" and "Prob", respectively. It can be observed that the results almost overlap when $\gamma_0 = 0$ and 5 dB, and also accurate when γ_0 is smaller, say $\gamma_0 = -5$ dB. This suggests that the special-case analysis is also applicable for more general case.

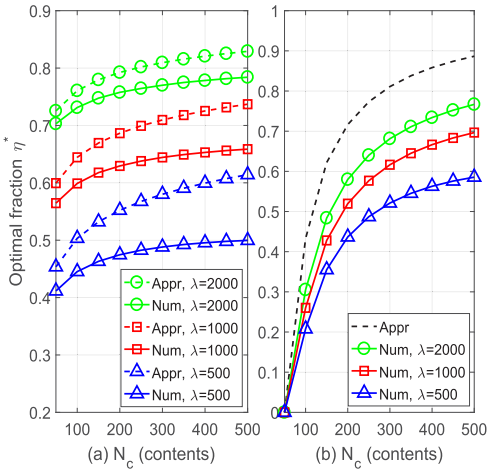


FIGURE 4. Accuracy of the approximations of the closed-form solution. (a) $\beta = 1.5$, (b) $\beta = 2.5$ ($\gamma_0 = 0$ dB).

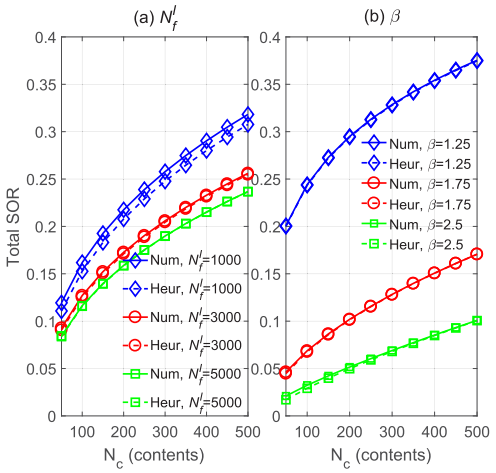


FIGURE 5. Performance of hybrid caching with heuristic cache resource allocation.

In Fig. 4, we provide the approximated results of the closed-form solution in (18) with $\beta = 1.5$ and 2.5 (with legend "Appr") and the numerical results obtained by (12) (with legend "Num"). It is shown that the approximation error increases with $\frac{N_c}{N_f^S}$ (i.e., $\frac{N_c}{\lambda T}$) as indicated in Appendix C. Despite that the approximation is not accurate, one can find that the approximated and numerical results exhibit the same trends for λ. The same trends can be also observed for ρ_0 and $\mathbb{E}[V_m]$, but the results are not shown for conciseness. This indicates that the insights gained in *Remark 3* are valid.

2) A HEURISTIC CACHE RESOURCE ALLOCATION METHOD

Recall from *Remark 3* that the fraction of the average number of requests for the SNM contents w^S exhibits the same monotonic properties as η^* in terms of ρ_0, λ and $\mathbb{E}[V_m]$. To avoid learning every traffic parameter, we heuristically set the optimal fraction η^* as w^S, by ignoring the impacts of N_c, N_f^I and β despite that η^* also depends on these three parameters as indicated in (18), which incurs marginal performance loss as shown in the following results.

In Fig. 5, we show the impacts of N_c, N_f^I and β on the performance of hybrid caching with the heuristic allocation method (i.e., $\eta^* \approx w^S$, with legend "Heur") by comparing with the optimal allocation with legend "Num", which is numerically obtained by (12a), and the total SOR is computed by (10). It is shown that the hybrid caching with the heuristic allocation has a performance loss from the numerically optimal allocation when $N_f^I = 1000$ contents, but the performance loss is negligible. N_c and β also have marginal impacts on the performance loss. Therefore, despite that η^* depends on N_c, N_f^I and β while w^S does not, we can safely use w^S as an accurate approximation of η^*. In other words, the heuristic cache resource allocation, $\eta^* \approx w^S$, is able to perform well.

IV. LEARNING BASED HYBRID CACHING

To implement the hybrid caching in practice, the popularity distribution and the fraction of the cache resource allocation in the next caching update duration need to be learned from the historical data of requests. In this section, we adopt a widely-used neural network, multilayer perceptron (MLP) [32], as an illustrative technique to make the prediction.

A. PREDICTION OF POPULARITY DISTRIBUTION

In practice, the content catalogue varies with time, and the popularity distribution is non-stationary that is hard to predict directly. To circumvent this difficulty, we first predict the number of requests in the next caching update duration (called popularity for short in the following) for each content, and then estimate the popularity distribution in the duration.

We set the length of the observation window as T_o times of the update duration. For example, if the caching update duration T_u is one day, then the observation window duration is T_o days.

We employ MLP to predict the popularity of each content in the next update duration by using the number of requests in the observation window, as shown in Fig. 6. In particular, each sample to train and test the MLP consists of a T_o-dimension input vector $\mathbf{x}$ and the output y. The input vector $\mathbf{x}$ for the f-th content is $\mathbf{x} = [r_f^1, \cdots, r_f^{T_o}]$, where r_f^t denotes the number of requests for the f-th content recorded in the t-th update duration of the observation window, and the output is the popularity in the next update duration, i.e., $y = r_f^{T_o+1}$.

After trying several activation functions such as tanh, sigmoid and ReLU functions [32], we select ReLU function (i.e., $y = \max(x, 0)$) as the activation function for hidden layers, which performs the best according to our simulation. To yield the non-negative popularity, we select the softplus function (i.e., $y = \log(1 + e^x)$), an activation function commonly applied for regression problem [32], in the output layer, again after trying several activation functions.

The MLP is trained to minimize the mean square error between the output and expected output as well as a regularization term with coefficient ν to reduce overfitting,

i.e.,

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} (y^{(n)} - \hat{y}^{(n)})^2 + \frac{\nu}{2} \|\mathbf{W}\|_F^2, \qquad (19)$$

where N_{tr} is the number of training samples, $\mathbf{W}$ is the weighting matrix between layers and $\mathbf{b}$ is the bias vector, and $y^{(n)}$ and $\hat{y}^{(n)}$ are the expected output and the output of the MLP with input $\mathbf{x}^{(n)}$, respectively. The backpropagation algorithm is adopted with the iterative batch gradient descent optimization [32], and the learning rate is adaptively adjusted with Adam algorithm [33]. To further reduce overfitting, we also employ the early stopping technique [32].

After learning the popularities of N_o contents in the observation window, i.e., $\hat{y}^{(1)}, \cdots, \hat{y}^{(N_o)}$, we can estimate the popularity distribution by normalizing the predicted popularities as $\hat{p}^f_{T_o+1} = \frac{\hat{y}^{(f)}}{\sum_{n=1}^{N_o} \hat{y}^{(n)}}, f = 1, \cdots, N_o$.

B. PREDICTION OF THE ALLOCATION FRACTION OF CACHE RESOURCE

Since the popularities of the new contents (i.e., the contents that are not requested in the observation window) cannot be predicted, these contents have to be cached in reactive manner and hence can be regarded as the SNM contents. Since unpopular contents contribute little to the performance of reactive caching, we only consider the popular contents with the numbers of requests in each update duration no less than a given value. To obtain the cache resource allocated for reactive caching by learning from the historical data, we consider the heuristic cache resource allocation method.

We still use MLP to predict w^S in the next update duration using the past values of w^S. The input and output of a sample are denoted as $\mathbf{x}' = [w^S_1, \cdots, w^S_{T_o}]$ and $y' = w^S_{T_o+1}$, respectively, where w^S_t is the ratio of the number of requests for the contents that are not requested in the observation window but are requested in the t-th update duration in $\mathcal{F}_t$ to the total number of requests for all contents in $\mathcal{F}_t$. Here, $\mathcal{F}_t$ is the set of contents in the t-th update duration, whose popularities are no less than a popularity threshold r_{th}.

The structure of MLP is similar to that in Fig. 6, but ReLU function is adopted as the activation function of the output layer, since it outperforms the softplus function for learning w^S according to simulations. The training process, learning algorithm and the techniques of reducing overfitting are the same as those for predicting the popularity with MLP.

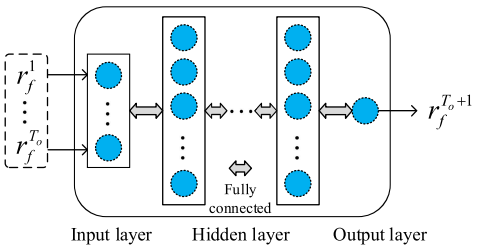


FIGURE 6. Structure of the MLP to predict the popularity (i.e., the number of requests in T_u for a content).

C. IMPLEMENTATION OF THE LEARNING BASED HYBRID CACHING POLICY

The procedure to implement the learning based hybrid proactive and reactive caching policy is summarized as follows.

In the *content placement phase*, i.e., at the beginning of the next update duration, each BS first divides its cache resource into two parts for proactive caching and reactive caching according to the predicted w^S in Sec. IV-B. Then, the popularity distribution for the not-new-contents of the next update duration is predicted as $\hat{p}^f_{T_o+1}, f = 1, \cdots, N_o$ using the method in Sec. IV-A. Next, the probabilistic caching policy is optimized with (12b) based on the predicted popularity distribution. Finally, each BS independently caches contents in the proactively managed storage according to the optimized caching probabilities.

In the *content delivery phase*, when a user requests a content, the system will first check whether the content has been cached at the BSs satisfying the SIR threshold. If the content can be found at the BSs meanwhile satisfying the SIR threshold, the nearest BS that caches the content will serve the user directly from its local cache. Otherwise, the nearest BS to the user will serve the user by fetching the requested content from the core network via backhaul, and meanwhile updates its reactively cached contents according to the LRU policy when the requested content is not cached at the proactively managed storage of any BS.

V. PERFORMANCE EVALUATION WITH SYNTHETIC AND REAL DATASETS

In this section, we compare the performance of the proposed hybrid proactive and reactive caching policy with non-hybrid proactive caching policy and reactive caching policy as well as the existing hybrid caching policy via both synthetic and real datasets.

A. DATASETS

1) REAL DATASETS

Two real datasets are collected from a university campus during the period from Oct. 29 to Nov. 16 in 2016 covering consecutive 80 days, which record requests from anonymous users to the videos from two famous content providers in China, i.e., *Youku* and *iQiYi*. Although the raw datas are not collected from the BSs, they can reflect the realistic traffic characteristics of video service in a small region, which is about 2 km^2. According to the statistics, for the two datasets, there are very few requests from 0:00 a.m. to 8:00 a.m., which have no impact on the performance evaluation of the caching policies. Therefore, we filter out the request data during this period. After the preprocessing, some statistical information is listed in TABLE 3.

TABLE 3. Statistics of *Youku* and *iQiYi* datasets.

Datasets	*Youku*	*iQiYi*
Number of requests	187, 216	111, 906
Number of videos	89, 724	41, 081
Number of average daily requested videos	1, 703	951
Average daily request rate for SNM contents	1, 507	793
Average daily request rate for IRM contents	728	572

2) SYNTHETIC DATASETS

The synthetic static and dynamic datasets are respectively synthesized with IRM and SNM models, also covering consecutive 80 days, the same as that in the real datasets. According to the statistics about the collected *Youku* dataset, about 40% of the most popular 100 videos are with different lifespans between 1 day and 20 days, and about 70% of them are with exponential popularity profile. Hence, we synthesize the *SNM* dataset by setting uniformly distributed lifespans in $[1, 20]$ days and the exponential popularity profile, i.e., $\Lambda_m(t) = 2/T_m e^{-2t/T_m}, t > 0$ [19]. The other traffic parameters of both the *IRM* and *SNM* datasets are set the same as those in TABLE 2.

B. SAMPLE GENERATION AND FINE-TUNED HYPER-PARAMETERS

1) PREDICTION OF POPULARITY DISTRIBUTION

Both the synthetic and real datasets consist of massive unpopular contents, e.g., 98.1% and 96.2% of all contents in the *Youku* and *iQiYi* datasets are requested less than 10 times over 80 days, respectively. The training samples generated from these unpopular contents contribute little to the prediction performance of MLP. Therefore, in the offline training phase, we remove the contents requested less than 10 times from both the synthetic and real datasets, and generate training and validation samples from the relatively popular contents. In the prediction phase, however, we evaluate the caching performance for all contents in the non-filtered datasets.

The first 70 days are used for training and validation. By sliding the observation window one update duration each time, the training and validation samples can be successively generated. Taking the case $T_\mathrm{u} = 1$ day and $T_\mathrm{o} = 5$ as an example, the samples with the expected outputs from 6-th day to 60-th day are taken as the training set, which account for about 85% of the total number of samples, and the remaining samples with the expected outputs from 61-st day to 70-th day are set as the validation set for fine-tuning the hyper-parameters. Since the generated samples for contents without any request in the observation window cannot be used for predicting the popularity, we remove these invalid samples from the training and validation sets.

After fine-tuning, the MLP models for all datasets are with one hidden layer, which contains 50 nodes. The learning rate of Adam algorithm is 10^{-4}. The batch size is 512 and the regularization coefficient is 0.03.

2) PREDICTION OF ALLOCATION FRACTION OF CACHE RESOURCE

Given the popularity threshold r_th, we can learn w^S in an update duration as described in Sec. IV-B. We also use the request data in the first 70 days for training and validation. For example, when $T_\mathrm{u} = 1$ day and $T_\mathrm{o} = 5$, we can compute w^S from the 6-th day to the 70-th day and obtain totally 65 allocation fractions. By sliding the observation window one update duration each time, 60 samples can be generated in turn. We take the first 50 samples with the expected outputs from the 11-st day to the 60-th day as the training set, and the remaining samples with expected outputs from the 61-st day to the 70-th day as the validation set.

After fine-tuning, the batch size is 2 for the MLP models to predict the allocation fraction, and the other hyper-parameters are the same as those in the MLP model to predict the content popularity.

C. PERFORMANCE COMPARISON

In this subsection, we evaluate the performance of the proposed hybrid caching policy (with legend " *P-Hybrid*") by comparing with the following baselines (respectively with legends "*LRU*", " *MLP*" and "*E-Hybrid*") using both synthetic and real datasets.

- *LRU:* This policy keeps track of the recent request time for the cached contents, and when the cache is full, it replaces the least recently requested content with the newly requested content to satisfy the cache resource constraint of each BS.
- *MLP:* This is a purely proactive caching policy. The policy is obtained by first predicting the popularity distribution by MLP in Sec. IV-A and then optimizing the probabilistic caching policy using (12b). With the optimized caching policy, each BS independently caches contents in the proactive manner.
- *E-Hybrid:*[1] This is a modified version of the existing hybrid proactive and reactive caching policy proposed in [17], where the prediction model, proactive policy and reactive policy are the same as our hybrid policy for a fair comparison. This policy first predicts the popularity distribution by the MLP in Sec. IV-A, then caches contents in each BS according to the optimized caching policy to initialize the whole cache list, and finally the cache space is updated by LRU during the next update duration.

We adopt the last 10 days for all datasets as the test period to evaluate the caching performance. The metric is the total SOR, i.e., the ratio of the overall successfully offloaded requests to all requests during the test period, where the SIR distribution for each request is numerically computed by $\mathbb{P}\{\gamma_f > \gamma_0\}$ in (5) for the proactively cached content or $\mathbb{P}\{\gamma > \gamma_0\}$ in (6) for the reactively cached content.

[1] Due to the optimization of cache resource allocation, the proposed hybrid caching policy outperforms the hybrid policy in [18] undoubtedly, and hence we omit the simulation results of the policy in [18].

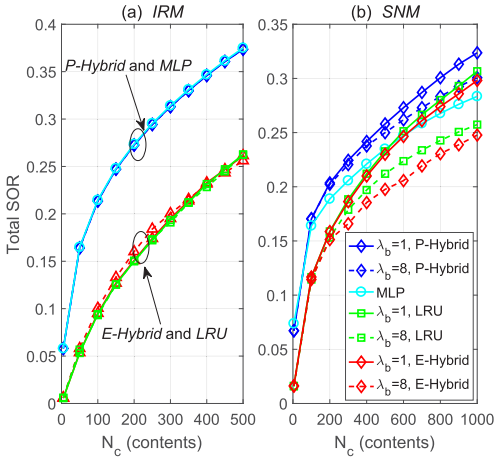


FIGURE 7. Performance comparison via synthetic datasets, a) IRM, b) SNM ($\gamma_0 = -10$ dB, $r_{\text{th}} = 3$, $T_{\text{u}} = 1$ day and $T_{\text{o}} = 5$).

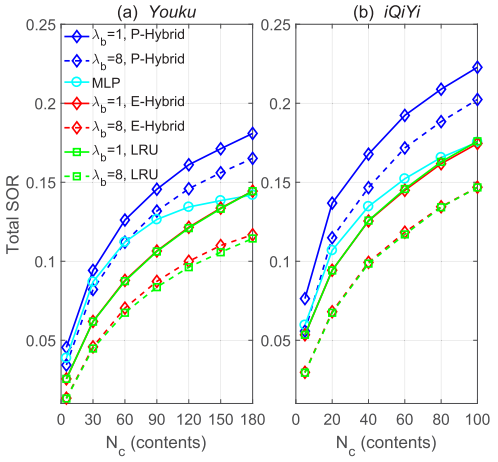


FIGURE 8. Performance comparison via real datasets, a) *Youku*, b) *iQiYi* ($\gamma_0 = -10$ dB, $r_{\text{th}} = 3$, $T_{\text{u}} = 1$ day and $T_{\text{o}} = 5$).

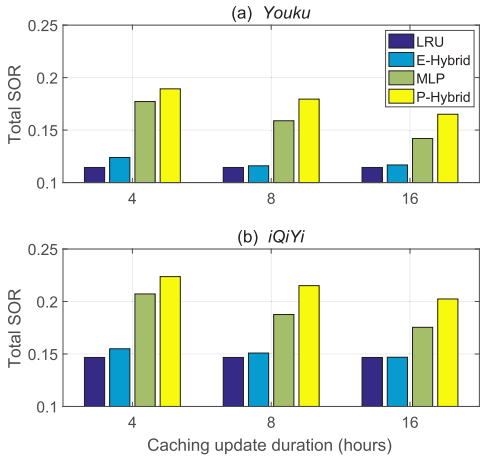


FIGURE 9. Impact of caching update duration ($\lambda_{\text{b}} = 8$, $N_{\text{c}} = 180$ contents for *Youku* dataset, and $N_{\text{c}} = 100$ contents for *iQiYi* dataset, $\gamma_0 = -10$ dB, $r_{\text{th}} = 3$, and the length of observation window is five days).

To show the impact of the density of the randomly-deployed BSs on the SOR, we assume that the requests are uniformly generated from users in each cell of a 2 km^2 region, considering that all the datasets do not contain location information of the requests. The cache size of each BS is constant no matter if the BSs are sparsely (i.e., $\lambda_{\text{b}} = 1$ in the legends) or densely (i.e., $\lambda_{\text{b}} = 8$ in the legends) deployed at this region.

In Fig. 7, we show the performance of the four caching policies on the synthetic *IRM* and *SNM* datasets, where the maximal cache sizes for *IRM* and *SNM* datasets are respectively 500 and 1000 contents, corresponding to about 10% of the catalogue sizes of both datasets.

It can be observed from Fig. 7(a) that the proposed hybrid caching policy performs the same as the proactive caching policy (i.e., *MLP*), and outperforms *LRU* and *E-Hybrid*. This is because in the synthetic *IRM* dataset, the popularity is static without newly-arrived contents, hence all cache resources are allocated to proactive caching.

It can be observed from Fig. 7(b) that the proposed hybrid caching policy outperforms *MLP*, *LRU* and *E-Hybrid* under the same BS density. This is because in the synthetic *SNM* dataset, there are always new contents being requested in the next update duration, hence we need to reserve some cache resource for reactive caching. It can be also observed that with the increasing density of the BSs, the performance of both *LRU* and the hybrid caching policies reduces, but the proposed hybrid caching policy still performs the best under the same density. This is because as the density of the BSs increases, the number of requests received at each BS decreases, which will deteriorate the performance of *LRU*. Since the proactive caching policy is optimized according to the predicted popularity distribution of this region and $\mathbb{P}\{\gamma_f > \gamma_0\}$ is irrelevant to the density of BSs, the performance of the proactive caching is not affected by the density of BSs.

In Fig. 8, we show the performance of the proposed hybrid caching policy evaluated with the two real datasets, where the maximal cache sizes for *Youku* and *iQiYi* are respectively 180 and 100 contents, corresponding to about 10% of the average numbers of daily requested contents as shown in TABLE 3. We can see that the proposed hybrid caching policy outperforms the existing hybrid caching policy, LRU and the purely proactive caching policy. The gain of the proposed hybrid caching policy over *E-Hybrid* comes from avoiding the eviction of popular contents by caching them in the proactively managed storage. The gain over *LRU* comes from predicting the popularity of the not-new contents. The gain over *MLP* comes from the quick response to the bursty popularity by caching the newly-arrived contents that become popular rapidly.

In Fig. 9, we show the impact of the caching update duration T_{u} on the performance of caching. Since the length of the observation window was defined as multiple times of T_{u}, the number of requests in the observation window will be too few to make the prediction if T_{u} is small. To observe the impact of T_{u}, we ensure that the observed numbers of requests are identical for different values of T_{u} by setting

the length of the observation window as five days. Then, for a short update duration, say four hours, we can obtain $T_o = 20$. It is shown from the results that the performance of *MLP* and *P-Hybrid* improves with the decrease of the update duration. This is because by reducing the update duration, the proactive caching policy can track the time-varying popularity more quickly. We can also observe that with shorter update duration, e.g., four hours, *MLP* exhibits larger gain over *LRU* by predicting the popularity quickly and caching the newly-arrived popular contents in time.

VI. CONCLUSION

In this paper, we optimized a hybrid proactive and reactive caching policy by taking into account the dynamic popularity in cache-enabled wireless networks, where the probabilistic caching policy and LRU were employed to cache contents that can be respectively modeled by IRM and SNM. Aimed at maximizing the total successful offloading ratio by caching both IRM and SNM contents at BSs, we formulated and solved an optimization problem of the hybrid caching policy including both the cache resource allocation among the two classes of contents and the probability of caching each IRM content. To provide the guideline to design hybrid caching policy without the assumption of knowing various traffic parameters, we further derived a closed-form solution of the optimal allocation fraction of cache resource under a special case. We proposed a heuristic but effective method to obtain the optimal allocation fraction, and predicted the popularity distribution as well as the allocation fraction with neural networks by using historical data. We evaluated the performance of the proposed hybrid caching policy via both synthetic and real datasets. The results demonstrated obvious gains of the proposed hybrid caching policy over existing hybrid proactive and reactive caching policy and non-hybrid policies with the real datasets.

APPENDIX A
PROOF OF PROPOSITION 1

Under the small-cache scenario, we have $\frac{N_c}{N_f^S} = \frac{N_c}{\lambda T} \approx 0$, and hence $\frac{\eta N_c}{\lambda T} \approx 0$ for $0 \le \eta \le 1$. Then, we can obtain from (8) that

$$\frac{1}{T}\int_0^{\infty} 1 - \phi_{V_m}\left(-\int_0^{t_c} \Lambda(\tau-\theta)\mathrm{d}\theta\right)\mathrm{d}\tau \approx 0. \quad \text{(A.1)}$$

Considering that the left-hand side of (A.1) monotonically increases with t_c and equals to zero when $t_c = 0$, we know that to make equation (A.1) hold, t_c should approximate to zero. With $t_c \approx 0$, we can further derive that $\int_0^{t_c} \Lambda(\tau-\theta)\mathrm{d}\theta \approx \Lambda(\tau)t_c$, and that $\phi_{V_m}(-\Lambda(\tau)t_c)$ can be approximated as $1 - \Lambda(\tau)t_c\mathbb{E}[V_m]$ by the first order Taylor expansion for $\Lambda(\tau)t_c \approx 0$.

Then, (8) can be approximated as

$$\eta N_c \approx \lambda t_c \mathbb{E}[V_m]\int_0^{\infty}\Lambda(\tau)\mathrm{d}\tau = \lambda t_c \mathbb{E}[V_m]. \quad \text{(A.2)}$$

Finally, by substituting (A.2) into (7) and considering $\int_0^{t_c}\Lambda(\tau-\theta)\mathrm{d}\theta \approx \Lambda(\tau)t_c$, (9) is derived.

APPENDIX B

The KKT conditions of problem **P1** can be derived as

$$-\frac{g_s(\eta)N_c}{1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1)} + \mu N_c + \lambda_1 - \nu_1 = 0, \quad \text{(B.1a)}$$

$$-g_i(c_f, q_f) + \mu + \lambda_2^f - \nu_2^f = 0, \quad \forall f \quad \text{(B.1b)}$$

$$\mu(\eta N_c + \|\mathbf{c}\|_1 - N_c) = 0, \quad \text{(B.1c)}$$

$$\lambda_1(\eta - 1) = 0, \quad \nu_1\eta = 0, \quad \text{(B.1d)}$$

$$\lambda_2^f(c_f - 1) = 0, \nu_2^f c_f = 0, \quad \forall f \quad \text{(B.1e)}$$

$$\text{(11b), (11c), (11d)}, \mu, \lambda_1, \nu_1, \lambda_2^f, \nu_2^f \ge 0, \quad \forall f \quad \text{(B.1f)}$$

where $g_s(\eta) \triangleq \frac{w^S}{N_f^S\mathbb{E}^2[V_m]}\mathbb{E}\{V_m^2\, e^{-\frac{V_m\eta N_c}{N_f^S\mathbb{E}[V_m]}}\}$ monotonically decreases with η, and $g_i(c_f, q_f) \triangleq \frac{\gamma_0^{2/\alpha}\epsilon_0 w^I q_f}{((1-\gamma_0^{2/\alpha}\epsilon_1)c_f + \gamma_0^{2/\alpha}\epsilon_0)^2}$.

To find the optimal solution from the KKT conditions, we first prove $\mu^* > 0$.

Assume that $\mu^* = 0$. Since $g_i(c_f^*, q_f) > 0$ and $\nu_2^{f*} \ge 0$, we have $\lambda_2^{f*} > 0, \forall f$ according to (B.1b). Then, $c_f^* = 1, \forall f$ can be obtained from (B.1e). Similarly, $\eta^* = 1$ can be obtained from (B.1a) and (B.1d), which violates the constraint in (11d). Therefore, $\mu^* > 0$.

Then, (B.1c) can be reduced to

$$\eta^* N_c + \|\mathbf{c}^*\|_1 = N_c. \quad \text{(B.2)}$$

When $g_s^{-1}[\mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))] > 1$, we have $g_s(\eta^*) \ge g_s(1) > \mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))$. In this case, $\lambda_1^* > 0$ according to (B.1a) and $\eta^* = 1$ according to (B.1d). When $g_s^{-1}[\mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))] = 1$, $g_s(\eta^*) > \mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))$ if $0 \le \eta^* < 1$, which however results in $\eta^* = 1$ according to (B.1a) and (B.1d), contradicting to $0 \le \eta^* < 1$. Therefore, $\eta^* = 1$.

When $g_s^{-1}[\mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))] < 0$, we have $g_s(\eta^*) \le g_s(0) < \mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))$. In this case, $\nu_1^* > 0$ according to (B.1a) and $\eta^* = 0$ according to (B.1d). When $g_s^{-1}[\mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))] = 0$, $g_s(\eta^*) < \mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))$ if $0 < \eta^* \le 1$, which however results in $\eta^* = 0$ contradicting to $0 < \eta^* \le 1$. Therefore, $\eta^* = 0$.

When $0 < g_s^{-1}[\mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))] < 1$, we have $g_s(1) < \mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1)) < g_s(0)$. In this case, if $\eta^* = 0$, then $\lambda_1^* > 0$ according to (B.1a) which results in $\eta^* = 1$, contradicting to $\eta^* = 0$. Similarly, if $\eta^* = 1$, then $\nu_1^* > 0$ according to (B.1a), which results in $\eta^* = 0$, contradicting to $\eta^* = 1$. Therefore, $0 < \eta^* < 1$ and $\lambda_1^*, \nu_1^* = 0$ according to (B.1d). We further have $\eta^* = g_s^{-1}[\mu^*(1+\gamma_0^{2/\alpha}(\epsilon_0-\epsilon_1))]$ according to (B.1a).

With similar derivations, we can obtain the optimal caching probability as $c_f^* = \left[\frac{1}{1-\gamma_0^{2/\alpha}\epsilon_1}\left(\left(\frac{\gamma_0^{2/\alpha}\epsilon_0 w^I q_f}{\mu}\right)^{\frac{1}{2}} - \gamma_0^{2/\alpha}\epsilon_0\right)\right]_0^1, \forall f$. Finally, according to the solution of η^* and $\mathbf{c}^*$, μ^* can be solved from (B.2).

APPENDIX C
PROOF OF PROPOSITION 2

With $f_V(v) = \frac{\beta V_{\min}^{\beta}}{v^{\beta+1}}$, $g_l(\eta^*)$ can be expressed as

$$g_l(\eta^*) = \frac{\beta^2 V_{\min}^{\beta}}{(\beta-1)\mathbb{E}[V_m]T} \int_{V_{\min}}^{\infty} \frac{e^{-\frac{v\eta^* N_c}{N_f^S \mathbb{E}[V_m]}}}{v^{\beta-1}} \mathrm{d}v. \quad \text{(C.1)}$$

Considering that $\int_x^{\infty} \frac{e^{-av}}{v^b}\mathrm{d}v = a^{b-1}\Gamma(1-b, ax)$ for positive a, b and x, where $\Gamma(s, y)$ is the upper incomplete Gamma function, we can rewrite $g_l(\eta^*)$ by substituting $\mathbb{E}[V_m] = \frac{\beta V_{\min}}{\beta-1}$ into the left-hand side of (C.1) as

$$g_l(\eta^*) = \beta\left(\frac{\beta-1}{\beta}\right)^{\beta-1} \frac{\mathbb{E}[V_m]}{T} \left(\frac{\eta^* N_c}{N_f^S}\right)^{\beta-2} \Gamma\left(2-\beta, \frac{\eta^* N_c}{N_f^S}\frac{\beta-1}{\beta}\right). \quad \text{(C.2)}$$

For $1 < \beta < 2$, we can obtain that $\Gamma\left(2-\beta, \frac{\eta^* N_c}{N_f^S}\frac{\beta-1}{\beta}\right) \approx \Gamma(2-\beta)$ in the small-cache scenario where $\frac{N_c}{N_f^S} \approx 0$, which leads to the result in (C.3). For $\beta > 2$, given that $\Gamma(s, y) \approx -y^s/s$ when $y \approx 0$ and $s < 0$, $g_l(\eta^*)$ can be approximated as (C.4). In summary, we can obtain

$$g_l(\eta^*) \approx \begin{cases} \frac{A_\beta \mathbb{E}[V_m]}{T}\left(\frac{\eta^* N_c}{N_f^S}\right)^{\beta-2}, & \text{if } 1 < \beta < 2, \quad \text{(C.3)} \\ \frac{\beta-1}{\beta-2}\frac{\mathbb{E}[V_m]}{T}, & \text{if } \beta > 2, \quad \text{(C.4)} \end{cases}$$

where $A_\beta = \beta\left(\frac{\beta-1}{\beta}\right)^{\beta-1}\Gamma(2-\beta)$.

The approximations of $g_l(\eta^*)$ in (C.3) and (C.4) are obtained based on $\frac{N_c}{N_f^S} \approx 0$. Therefore, they are more accurate when $\frac{N_c}{N_f^S}$ decreases.

By substituting (C.3) and (C.4) into (17), we obtain

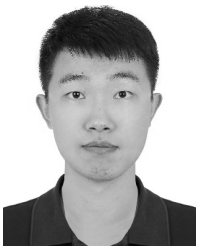
$$\eta^* \approx \begin{cases} h^{-1}(DN_c), & \text{if } 1 < \beta < 2, \\ \max\left\{1 - \left(\frac{\rho_0 T}{\mathbb{E}[V_m]}\frac{\beta-2}{\beta-1}\right)^{\beta} \frac{(N_f^I)^{1-\beta}}{N_c}, 0\right\}, & \text{if } \beta > 2, \end{cases} \quad \text{(C.5)}$$

where $D \triangleq \left(\left(\frac{A_\beta \mathbb{E}[V_m]}{\rho_0 T}\right)^{\beta}(N_f^I)^{\beta-1}\left(\lambda T\right)^{(2-\beta)\beta}\right)^{\frac{1}{(\beta-1)^2}} > 0$ for $1 < \beta < 2$, $h(x) = x^{\frac{(2-\beta)\beta}{(\beta-1)^2}}/(1-x)^{\frac{1}{(\beta-1)^2}}$, $h^{-1}(\cdot)$ is the inverse function of $h(\cdot)$. For $1 < \beta < 2$, it is easy to show that $h(\eta^*)$ monotonically increases with η^* by examining its first derivative. Thus, $h^{-1}(DN_c)$ monotonically increases with N_c when $D > 0$.

REFERENCES

[1] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.
[2] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.
[3] E. Baştuğ, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data meets telcos: A proactive caching perspective," *J. Commun. Netw.*, vol. 17, no. 6, pp. 549–557, Dec. 2015.
[4] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
[5] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5g wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2995–3007, Apr. 2016.
[6] Q. Li, W. Shi, X. Ge, and Z. Niu, "Cooperative edge caching in software-defined hyper-cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2596–2605, Nov. 2017.
[7] J. Wen, K. Huang, S. Yang, and V. O. K. Li, "Cache-enabled heterogeneous cellular networks: Optimal tier-level content placement," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5939–5952, Sep. 2017.
[8] S. Kuang, X. Liu, and N. Liu, "Analysis and optimization of random caching in K-tier multi-antenna multi-user HetNets," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5721–5735, Aug. 2019.
[9] D. Liu and C. Yang, "Caching policy toward maximal success probability and area spectral efficiency of cache-enabled HetNets," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2699–2714, Jun. 2017.
[10] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *Proc. IEEE ITC*, Sep. 2012, pp. 1–8.
[11] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16–22, Aug. 2016.
[12] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1111–1125, Jun. 2018.
[13] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2017.
[14] N. Zhang, K. Zheng, and M. Tao, "Using grouped linear prediction and accelerated reinforcement learning for online content caching," in *Proc. IEEE ICC Workshops*, May 2018, pp. 1–6.
[15] C. Zhong, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *Proc. IEEE CISS*, Mar. 2018, pp. 1–6.
[16] H. Zhu, Y. Cao, W. Wang, T. Jiang, and S. Jin, "Deep reinforcement learning for mobile edge caching: Review, new features, and open issues," *IEEE Netw.*, vol. 32, no. 6, pp. 50–57, Nov. 2018.
[17] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach," *IEEE Access*, vol. 5, pp. 5870–5881, 2017.
[18] C. Koch, S. Werner, A. Rizk, and R. Steinmetz, "MIRA: Proactive music video caching using ConvNet-based classification and multivariate popularity prediction," in *Proc. IEEE MASCOTS*, Sep. 2018, pp. 109–115.
[19] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Unravelling the impact of temporal and geographical locality in content caching systems," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1839–1854, Oct. 2015.
[20] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, "Placing dynamic content in caches with small population," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.
[21] Y. Tan, Y. Yuan, T. Yang, and B. Hu, "Learning-based caching with unknown popularity in wireless video networks," in *Proc. IEEE VTC Spring*, Jun. 2017, pp. 1–5.
[22] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proc. IEEE ICC*, Jun. 2015, pp. 3358–3363.
[23] Y. Cui, D. Jiang, and Y. Wu, "Analysis and optimization of caching and multicasting in large-scale cache-enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5101–5112, Jul. 2016.
[24] B. Chen, C. Yang, and Z. Xiong, "Optimal caching and scheduling for cache-enabled D2D communications," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1155–1158, Jan. 2017.
[25] B. Chen, C. Yang, and G. Wang, "High-throughput opportunistic cooperative device-to-device communications with caching," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7527–7539, Aug. 2017.
[26] F. Olmos, B. Kauffmann, A. Simonian, and Y. Carlinet, "Catalog dynamics: Impact of content publishing and perishing on the performance of a LRU cache," in *Proc. IEEE ITC*, Sep. 2014, pp. 1–9.
[27] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 126–134.
[28] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemp. Phys.*, vol. 46, no. 5, pp. 323–351, Dec. 2005.

[29] M. Garetto, E. Leonardi, and S. Traverso, "Efficient analysis of caching strategies under dynamic content popularity," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 2263–2271.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[31] D. Liu and C. Yang, "Optimal content placement for offloading in cache-enabled heterogeneous wireless networks," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1–6.
[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, to be published.



KAIQIANG QI received the B.S. degree in electronics engineering from Beihang University (formerly the Beijing University of Aeronautics and Astronautics), China, in 2015, where he is currently pursuing the Ph.D. degree in information and communication engineering with the School of Electronics and Information Engineering. His research interests lie in the area of wireless edge caching based on mobile big data and machine learning.

SHENGQIAN HAN received the B.S. and Ph.D. degrees from Beihang University, Beijing, China, in 2004 and 2010, respectively, where he is currently an Associate Professor with the School of Electronics and Information Engineering. From 2015 to 2016, he was a Visiting Scholar with the Department of Electrical Engineering, University of Southern California, Los Angeles, USA. He is also with the Hangzhou Innovation Institute, Beihang University, Hangzhou. His recent research interests include wireless big data, full-duplex network, and energy-efficient transmission. He has served as a Technical Program Committee Member of numerous IEEE conferences. He is an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*.

CHENYANG YANG received the Ph.D. degree in electrical engineering from Beihang University, China, in 1997, where she has been a Full Professor with the School of Electronics and Information Engineering, since 1999. She has published over 200 papers and filed over 80 patents in the fields of energy-efficient transmission, wireless local caching, URLLC, CoMP, interference management, cognitive radio, and relay. She was supported by the First Teaching and Research Award Program for Outstanding Young Teachers of Higher Education Institutions, Ministry of Education, China. Her recent research interests lie in wireless caching, resource allocation with machine learning, and URLLC. She was the Chair of the Beijing Chapter of the IEEE Communications Society, from 2008 to 2012, and the MDC Chair of APB of the IEEE Communications Society, from 2011 to 2013. She has served as a TPC Co-Chair, a Track Co-Chair, and a TPC Member of many IEEE conferences. She has ever served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and a Guest Editor for the *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING* and the *IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS*.