A Reinforcement Learning-Based User-Assisted Caching Strategy for Dynamic Content Library in Small Cell Networks

Xinruo Zhang, *Member, IEEE,* Gan Zheng, *Senior Member, IEEE,*
Sangarapillai Lambotharan, *Senior Member, IEEE,*
Mohammad Reza Nakhai, *Senior Member, IEEE,*
and Kai-Kit Wong, *Fellow, IEEE*

Abstract

This paper studies the problem of joint edge cache placement and content delivery in cache-enabled small cell networks in the presence of spatio-temporal content dynamics unknown *a priori*, where small base stations (SBSs) satisfy users' content requests either directly from their local caches, or by retrieving from other SBSs' caches or from the content server. In contrast to previous approaches that assume a static content library at the server, this paper considers a more realistic non-stationary content library, where new contents may emerge over time at different locations. To keep track of spatio-temporal content

Xinruo Zhang, Gan Zheng and Sangarapillai Lambotharan are with Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, LE11 3TU, U.K. (e-mail: {x.zhang, g.zheng, s.lambotharan}@lboro.ac.uk).

Mohammad Reza Nakhai is with Centre for Telecommunications Research, King's College London, WC2B 4BG, U.K. (e-mail: reza.nakhai@kcl.ac.uk).

Kai-Kit Wong is with the Department of Electronic and Electrical Engineering, University College London, London, WC1E 7JE, U.K.(e-mail: kai-kit.wong@ucl.ac.uk).

dynamics, we propose that the new contents cached at users can be exploited by the SBSs to timely update their flexible cache memories in addition to their routine off-peak main cache updates from the content server. To take into account the variations in traffic demands as well as the limited caching space at the SBSs, a user-assisted caching strategy is proposed based on reinforcement learning principles to progressively optimize the caching policy with the target of maximizing the weighted network utility in the long run. Simulation results verify the superior performance of the proposed caching strategy against various benchmark designs.

Index Terms

non-stationary bandit; cache placement; content delivery; time-varying popularity; dynamic content library

I. INTRODUCTION

Global mobile data traffic is growing at an unprecedented rate and is predicted to account for more than 63 percent of total data traffic, reaching 48.3 Exabytes per month by 2021 [1]. The content delivery network (CDN) that has been widely adopted for traffic congestion reduction, is expected to carry 71 percent of all internet traffic by 2021, of which 82 percent will be video traffic. However, the backhaul data rate demand between the base stations (BSs) and the core network incurred by such rapid traffic growth has become the major revenue and technical bottlenecks for the network operators, especially during peak traffic periods [3]. Due to the fact that a large portion of backhaul traffic is contributed by transmitting duplicate data from the core network to multiple users [4], caching popular contents, e.g., video, social media, news and maps, that are repeatedly requested by a large number of users in local memories installed at BSs to eliminate duplicate data transmission, has recently attracted significant attention of researchers [5]. The integration of content caching with small base stations (SBSs) that provide short-range and low-cost transmission underlaying the existing macrocell cellular networks, allows popular mobile data to be prefetched from the core network during off-peak traffic hours and to be delivered to edge users at peak times. Such integration provides opportunities not only to offload the backhaul traffic load, but also to improve system performance such as energy efficiency and transmission delay, and hence, significantly alleviates the backhaul and latency bottlenecks in conventional wireless CDN [6]. Considering the fact that the capacity of cache storage is highly limited at the individual SBSs as compared to the massive content library at the content server, efficient caching mechanisms are advocated to be developed for the network operators to maximally benefit from caching techniques. Recently, cooperative caching with joint optimization of different caching locations, e.g., central cloud caching and SBSs caching, has been proposed as a potential solution to the enhancement of content caching performance in dynamic mobile networks [3]. By coordinating content caching at different locations, the individual SBSs may cache differentiated contents and retrieve the requested content from other cache locations, rather than from the content server, at a lower cost. However, provided that the individual SBSs can only observe the instantaneous content requests of their users, the content popularity distribution and/or users' preference may be unknown *a priori* and may vary with time and locations. Hence, a timely estimation of users' content requests is challenging but essential for the effective caching policy design as well as for the reliable and cost-efficient operation of networks under the uncertainty of traffic demands.

A. Related Works

Most approaches in the literature assume finite cache storage with time-invariant content popularity distribution perfectly known at the BSs [5]–[15], and design either content placement strategies [6]–[11] or content delivery strategies [12]–[15] in various network scenarios. Joint consideration of content placement and delivery strategies have been studied in recent years based on either coded [16]–[19] or uncoded [20]–[23] data. The assumption of *a priori* knownledge of content popularity distribution, nevertheless, is not realistic in practical scenarios. In recent years, using machine learning techniques to predict the unknown content popularity, and proactively cache the popular contents at the BSs in advance of users' requests, has attracted the attention of the researchers [24]. In [25], the authors propose a Lyapunov optimization approach to hybrid content caching design to tackle spatial dynamics in traffic demands, where the content popularity is not required. The authors in [3] and [26] relax this assumption and introduce the multi-armed bandit (MAB) based learning approaches to estimate the content popularity distribution over time horizon. However, [3] only considers spatial diversity of the static content popularity, whilst [26] assumes unknown and time-invariant content popularity. The authors in [27] model the cache replacement problem as a Markov decision process and propose a Q-learning algorithm to trade-off the global and local popularity demands in heterogeneous networks. Assuming a Poisson request model, [28] develops a transfer learning based approach with a finite training time to improve the estimation of the content popularity in a heterogenous network based on a training set of ratings. The work in [28], nevertheless, deals with content caching only in a single BS for one period in time, while an online learning approach may be more suitable for the estimation of content popularity over the time horizon. The authors in [29] propose a regret learning based per-BS caching strategy to learn the spatio-temporal traffic demands and to capture the trade-off of the local and the global content popularity. However, the aforementioned works simply ignore the fact that the contents can be dynamic over time: new contents are constantly introduced to the content library and their popularity distribution may change over time. For instance, the popularity of some contents such as news vanishes within a limited time whilst others such as music and movies may attract sustained requests for a long period of time. Hence, those works without considering the dynamic content library in the nature of their designs, may not be able to catch up with the rapid variations of the content demands in practice. The authors in [30] propose an ON-OFF traffic model to capture the impact of dynamic contents on cache performance based on Che's approximation, whereas, they have sacrificed the key fact that the request processes at different caches are independent.

B. Contributions

This paper focuses on joint design of edge cache placement and content delivery in small cell networks. In contrast to the existing caching designs that assume stationary content library and/or time-invariant content popularity, we consider a non-stationary content library with spatiotemporal content dynamics unknown *a priori*. The novel contribution of this paper is the development of a reinforcement learning (RL) based user-assisted caching algorithm that aims to keep track of the spatio-temporal content dynamics and maximize an average weighted utility of the network in the long run. The main contributions of this paper are summarized as follows:

- We propose to exploit users' caches to improve caching performance at the SBSs during peak hours. This is inspired by the fact that some users may have cached new contents through other networks, for example wireless local area networks (WLAN). To be specific, a portion of the cache unit at each SBS is allocated as the flexible cache memory, which can be timely updated with the new contents cached at the users in addition to the routine off-peak main cache update from the content server.
- A user-assisted caching algorithm is proposed based on a non-stationary bandit model to adaptively track the spatio-temporal variations of users' content demands and sequentially optimize the content caching and delivery policies over a long time horizon.
- We introduce a three-phase procedure at different time scales for joint cache placement and content delivery in small cell networks. Phase I is the content delivery phase at the individual time slots, where the content demand of each user is satisfied from one of the caching locations with different serving rewards. Phase II is the SBSs' flexible cache update phase, where the flexible caches of the SBSs can be more frequently updated with users' cached new contents. Phase III is the SBSs' main cache update phase at off-peak times, where the cache units of the SBSs are updated from the content server.
- To take into account the limited caching space at the SBSs, content caching coordination

is employed among SBSs and a near-optimal constrained cross-entropy (CCE) method is adopted in Phase III to solve the cache placement optimization problem with low-complexity.

C. Organization and Notations

The rest of this paper is organized as follows. Section II introduces the system model. In section III, the joint cache placement and content delivery problem is formulated and then decomposed into RL assisted content placement optimization problems that can be solved via the near-optimal CCE method. In section IV, a non-stationary bandit-inspired user-assisted caching algorithm is proposed to cope with the spatio-temporal content dynamics. Numerical simulation results are presented and analyzed in section V. Finally, section VI concludes the paper.

Notations: Throughout the paper, w and $\mathbf{w}$, respectively, indicate a scalar w and a vector $\mathbf{w}$. $\mathbb{E}(\cdot)$ is the expected value, $\mathbb{B}^{n\times m}$ denotes the binary space of n-by-m matrices and $\mathbb{CN}(0,1)$ is the zero-mean complex Gaussian random variables with unit variance. $\|.\|_0$ is the l_0-norm indicating the number of non-zero entries in the vector, and $I_{\{.\}}$ is an indicator function that returns one if $\{.\}$ holds true and zero otherwise.

II. SYSTEM MODEL

A. System Scenario

As illustrated in Fig. 1, we consider a time-slotted downlink small cell network consists of N_b SBSs serving K users over a shared frequency band. Let us denote by $\mathcal{L}_b = \{1, \cdots, N_b\}$, $\mathcal{L}_u = \{1, \cdots, K\}$ and $\mathcal{T} = \{1, \cdots, T\}$, respectively, the index sets of the SBSs, the users and the discrete time slots. The individual SBSs have a circular coverage area with communication radius of R_b. Each user may identify and communicate with its neighboring SBSs, whilst only one SBS will serve the user. Let us denote by $\mathcal{L}_u^b = \{1, \cdots, K_b\}$ the index set of users associated with SBS b. Featuring cache units, the individual SBSs are connected with each other via inter-SBS links, to the central processing unit (CU) via capacity-limited fronthaul links and to the

TABLE I

NOTATION

Symbol	Definition
$\mathcal{L}_b$ and $\mathcal{L}_u$	Index sets of SBSs and users
$\mathcal{L}_u^b$	Index set of users associated with SBS b
$\mathcal{T}$	Index set of discrete time slots in Phase I
$\tau^{\text{[flex]}}$ and $\tau^{\text{[main]}}$	Respective time scales of Phase II and Phase III
$\mathcal{F}$	Index set of finite content library at the server
$\mathcal{F}^t$	Index set of current content library at time t
S_f	Size of content f
$N^{\text{[new]}}$	Number of new contents added to the content server in Phase II
$\{\Delta_u\}_{u\in\mathcal{L}_u}$	The spatial shifts of content popularity among individual users
$\{\theta_{u,f}^t\}_{f\in\mathcal{F}^t}$	The unknown content popularities of user u at time t
$\mathbf{d}_u^t$	Binary content demand vector of user u at time t
$d_{uf}^t \in \{0,1\}$	Whether the content f is requested by user u at time t
$\pi^{\text{[local]}}$	Gross gain per unit content size of an SBS serving users from its local cache
$\pi^{\text{[SBS]}}$	Gross gain per unit content size of an SBS serving users by fetching content from other SBSs' caches
$\pi^{\text{[server]}}$	Gross gain per unit content size of an SBS serving users by fetching content from the content server
$\kappa^{\text{[user]}}$	Per-unit average discount rate for users' uploading incentives offered by an SBS
M_b	Capacity of cache unit at SBS b with a portion ξM_b being allocated for flexible cache memory
$\mathbf{c}_p^t$	Content caching placement policy at time t
$c_{b,f}^t \in \{0,1\}$	Whether content f is cached at SBS b at time t
$\mathbf{c}_r^t$	Content retrieving policy at time t
$c_{b,b',f}^t \in \{0,1\}$	Whether content f is retrieved by SBS b from SBS b' at time t
$c_{b,s,f}^t \in \{0,1\}$	Whether content f is retrieved by SBS b from the content server at time t
$\mathbf{c}_u^t$	Content uploading policy at time t
$c_{b,u,f}^t \in \{0,1\}$	Whether content f is uploaded from user u to SBS b at time t
$G_{b,f}^{u,t}$	Net gain for SBS b serving user u with content f directly from its local cache at time t
$G_{b,s,f}^{u,t}$	Net gain for SBS b serving user u by fetching content f from the content server at time t
$G_{b,b',f}^{u,t}$	Net gain for SBS b serving user u by fetching content f from SBS b' at time t
$\mathcal{R}_b(d_{uf}^t)$	Instantaneous serving reward for SBS b serving user u with content f at time t
$\bar{\mathbf{\Upsilon}}^t = \{\bar{\upsilon}_{b,f}^t\}$	Estimated joint reward distribution of contents at individual SBSs at time t

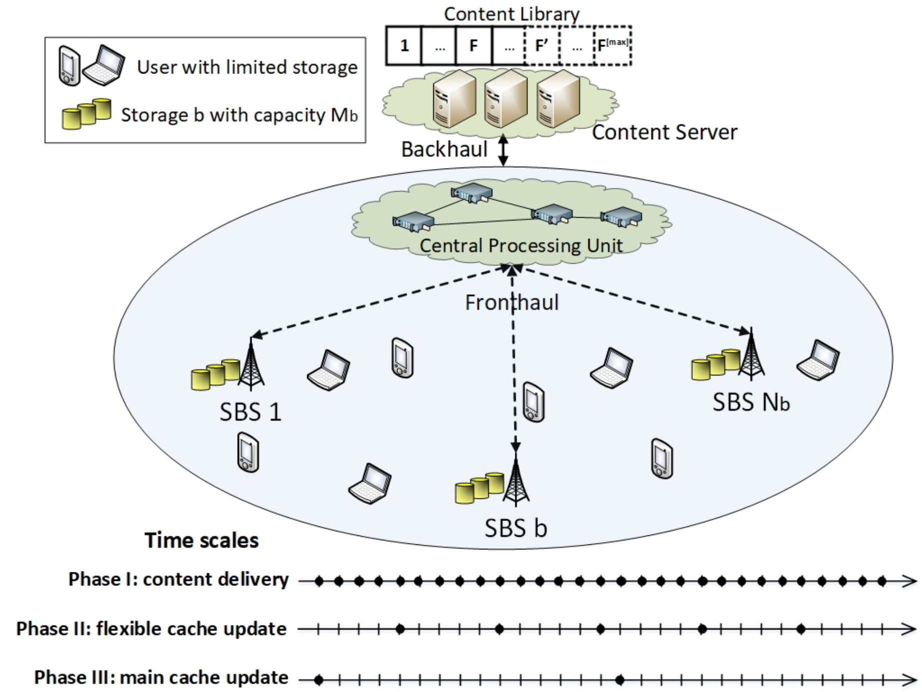

Content Library
1
...
F
...
F'
...
F[max]
User with limited storage
Storage b with capacity Mb
Content Server
Backhaul
Central Processing Unit
Fronthaul
SBS 1
SBS Nb
SBS b
Time scales
Phase I: content delivery
Phase II: flexible cache update
Phase III: main cache update

Fig. 1. Illustration of system scenario and three-phase procedure at different time scales for joint cache placement and content delivery.

content server via backhaul links. The CU coordinates all content caching and delivery strategies for the SBSs. A three-phase procedure at different time scales of $t \in \mathcal{T}$, $\tau^{[\text{flex}]}$ and $\tau^{[\text{main}]}$, is proposed for joint cache placement and content delivery, namely, Phase I: the content delivery phase; Phase II: the SBSs' flexible cache update phase; and Phase III: the SBSs' main cache update phase. The notations in this paper are listed in Table I.

1) Non-stationary Content Library: Let us consider a realistic scenario, where the new contents are constantly introduced into the system and the finite content library at the content server is thus non-stationary. Let us denote by $\mathcal{F} = \{1, \cdots, F, \cdots, F^{[\text{max}]}\}$ the finite content library with individual content sizes of $\{S_f\}_{f \in \mathcal{F}}$, where F and $F^{[\text{max}]}$, respectively, denote the initial and the maximum numbers of contents in the content library. Let us denote by $\mathcal{F}^t = \{1, \cdots, F'\}$ the content library at the t-th time slot, $t \in \mathcal{T}$, where F' indicates the current number of contents in the library and it is evident that $F \leq F' \leq F^{[\max]}$. Once the content library is full, i.e., $F' = F^{[\max]}$, the least recently used contents at the content server will be evicted and replaced by the newly emerged contents. Note that the content refreshment at the content server takes into consideration the content lifetime, and the dynamic content library adopted in the considered scenario naturally results in the time-varying content popularities, which are unknown *a priori*.

2) New Contents at Users: Each individual user is equipped with a capacity-limited local cache memory and can only cache one content at each time. At the end of time slot $t,\ t \in \mathcal{T}$, each user updates its cache memory with its requested content. In addition, the newly emerged contents may be cached at some random users either via being generated by the local users themselves, or by being brought in through other networks such as WLAN or due to users' mobility. The users are motivated to upload these potentially popular new contents to their neighboring SBSs for the incentive payments. The incentives for users to upload new contents can be earning extra data rate, extra bandwidth, and some discounts on their mobile data charges. Let us denote by $\kappa^{[\text{user}]}$ the per-unit average discount rate offered by an SBS for users' uploading incentives. For simplicity, an identical $\kappa^{[\text{user}]} \in [0,1]$ at all SBSs is assumed.

3) Users' Content Demands: We assume that the users' content request arrival processes are independent homogenous Poisson point processes with request rate of 1, i.e., each individual user on average may request one content that is not cached by itself from its neighboring SBSs at each time slot. Let the content demands of user $u,\ u \in \mathcal{L}_u$, at time t be denoted by $\mathbf{d}_u^t = \{d_{u1}^t, \cdots, d_{uf}^t, \cdots, d_{uF'}^t\}$, where the binary scalar $d_{uf}^t \in \{0,1\}$ indicates whether or not the content f is requested by user u at time t. Let us denote by $\{\theta_{u,f}^t\}_{f\in\mathcal{F}^t}$ the actual content popularities of user u that is unknown to the SBSs or the CU at time t. In addition to the temporal variability, we further consider the spatial diversity of the content popularity distributions among individual users, i.e., the users at different geographical locations may have diverse preferences for the contents. To this end, we model the spatial diversity by circularly shifting the content popularity distribution at user u by Δ_u with respect to user $u-1$. We further assume that the number of contents at the content library is usually much higher than the number of locally served users, hence the averaging effect over aggregated local users is an unlikely event in our considered caching problem. Without loss of generality, it is assumed that the longest delay for retrieving the largest content from the content server does not exceed a prescribed slot duration. If multiple content requests have been raised by a user at a time slot, those requests that can not be served within the given time slot will be dropped. The instantaneous content demands of the users at time slot t, i.e., $\{\mathbf{d}_u^t\}$, can be satisfied directly from the local cache of the serving SBS, or by retrieving from one of the caches of the other SBSs or from the content server with different gross gains per unit content size of $\pi^{[\text{local}]}$, $\pi^{[\text{SBS}]}$ and $\pi^{[\text{server}]}$, respectively. Given the fact that the corresponding latency is the longest for retrieving content from the content server while the shortest for fetching data from SBSs' local caches, the per-unit gross gains are set to be inversely proportional to the latency, i.e., $\pi^{[\text{server}]} \ll \pi^{[\text{SBS}]} < \pi^{[\text{local}]}$. For simplicity, let us assume identical $\pi^{[\text{local}]}$, $\pi^{[\text{SBS}]}$ and $\pi^{[\text{server}]} \in [0, 1]$ at all SBSs.

4) Cache Units at the SBSs: Each individual SBS is equipped with a cache unit with capacity of $M_b,\ b \in \mathcal{L}_b$. To fully exploit the contents cached at the local users, a portion with capacity of ξM_b of each cache unit is allocated as the flexible cache memory. The flexible cache memory that is made up of expensive and high-speed static random access memory (RAM), can be timely updated from the caches of the local users, whilst the remainder of the cache unit that is made up of cheaper and slower RAM, will be updated at a more infrequent pace, for instance, from the content server during off-peak traffic hours.

5) Three Phases of Different Time Scales: Recall that we consider a three-phase procedure at different time scales of $t \in \mathcal{T}$, $\tau^{[\text{flex}]}$ and $\tau^{[\text{main}]}$, for joint cache placement and content delivery.

- In Phase I, i.e., at each time slot $t,\ t \in \mathcal{T}$, the SBS associated with the highest serving reward will be chosen by the CU as the serving SBS. The individual serving SBSs then satisfy the instantaneous content requests of their scheduled users.
- In Phase II, i.e., for every $\tau^{[\text{flex}]}$ time slots, $N^{[\text{new}]}$ number of new contents are added to

the content server and might be cached by some random users. Each user broadcasts its cached content directory to its neighboring SBSs, and the CU will then make a decision on whether or not to update SBS's flexible cache with the user's cached content.

- In Phase III, i.e., for every $\tau^{\text{[main]}}$ time slots, the CU designs cache placement policy for the SBSs based on the reward information. The main cache replacements are executed accordingly from the content server to the SBSs via backhaul links.

B. Downlink Transmission

Let us denote by Ψ_{bu}^{t} the channel gain between SBS b and user u at the t-th time slot, $t \in \mathcal{T}$, and denote by $P_b^{\text{[Tx]}}$ the transmit power of SBS b. The signal-to-interference-plus-noise ratio (SINR) for user u served by SBS b at time slot t, $t \in \mathcal{T}$, can be expressed as

$$\text{SINR}_{bu}^{t} = \frac{P_b^{\text{[Tx]}}\Psi_{bu}^{t}}{\sum\limits_{b' \in \mathcal{L}_b, b' \neq b} P_{b'}^{\text{[Tx]}}\Psi_{b'u}^{t} + \sigma_u^2}, \tag{1}$$

where σ_u^2 is the variance of the additive white Gaussian noise at user u. With the normalized bandwidth, the instantaneous data rate for user u served by SBS b at time slot t, is given by

$$R_{bu}^{t} = \log_2(1 + \text{SINR}_{bu}^{t}). \tag{2}$$

C. Content Caching and Retrieving

Let us define the binary vector $\mathbf{c}_p^t = \{c_{b,f}^t \in \{0,1\},\ \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t\}$ as the content caching policy at time t, $t \in \mathcal{T}$, where $c_{b,f}^t = 1$ and $c_{b,f}^t = 0$ indicate that the content f is cached and is not cached at SBS b, respectively. This caching policy will be designed every $\tau^{\text{[main]}}$ time slots in Phase III for SBSs' main cache update, and, might be updated every $\tau^{\text{[flex]}}$ time slots in Phase II for the portion of flexible cache memory. Let us denote by $\mathbf{c}_r^t = \{c_{b,b',f}^t, c_{b,s,f}^t \in \{0,1\},\ \forall b' \neq b, b \in \mathcal{L}_b, b' \in \mathcal{L}_b, f \in \mathcal{F}^t\}$ the content retrieving policy at time t, where $c_{b,b',f}^t = 1$ and $c_{b,b',f}^t = 0$ indicate that the content f is fetched and is not fetched by SBS b from SBS b', respectively. $c_{b,s,f}^t \in \{0,1\}$ denotes whether or not the content f is retrieved from the content server by SBS

b at time t. The user demand of content $f,\ f \in \mathcal{F}^t$ at each time slot $t,\ t \in \mathcal{T}$, will either be satisfied by serving from the content server or from one of the SBSs, or be dropped, as

$$c^t_{b,s,f} + \sum_{b' \in \mathcal{L}_b} c^t_{b,b',f} \leq 1. \tag{3}$$

Let us denote the content uploading policy at time t as $\mathbf{c}^t_u = \{c^t_{b,u,f} \in \{0,1\},\ \forall b \in \mathcal{L}_b, u \in \mathcal{L}^b_u, f \in \mathcal{F}^t\}$, where $c^t_{b,u,f} = 1$ and $c^t_{b,u,f} = 0$, respectively, represent that the content f is uploaded and is not uploaded from user u to SBS b. Then, the net gain of SBS b storing content f and serving user u at time slot t, can be defined as

$$G^{u,t}_{b,f} = (\pi^{\text{[local]}} - \kappa^{\text{[user]}} \sum_{u' \in \mathcal{L}^b_u, u' \neq u} c^t_{b,u',f}) S_f d^t_{uf} c^t_{b,f},\ \forall t \in \mathcal{T}, u \in \mathcal{L}^b_u, b \in \mathcal{L}_b, f \in \mathcal{F}^t, \tag{4}$$

which indicates that the discount rate offered by SBS b need to be subtracted from the gross gain if SBS b serves user u with content f directly from its local cache and its cached content f is uploaded from the other local users. The net gain of SBS b for serving user u by retrieving content f from SBS b' at time slot t, is given by

$$G^{u,t}_{b,b',f} = (\pi^{\text{[SBS]}} - \kappa^{\text{[user]}} \sum_{u' \in \mathcal{L}^{b'}_u} c^t_{b',u',f}) S_f d^t_{uf} c^t_{b,b',f},\ \forall b' \neq b, b \in \mathcal{L}_b, b' \in \mathcal{L}_b, u \in \mathcal{L}^b_u, t \in \mathcal{T}, f \in \mathcal{F}^t, \tag{5}$$

which denotes that if the content f cached at SBS b' and retrieved by SBS b is uploaded from the local users of SBS b', the corresponding discount rate need to be subtracted. The net gain of SBS b for serving user u by retrieving content f from the content server at time slot t, is given by

$$G^{u,t}_{b,s,f} = \pi^{\text{[server]}} S_f d^t_{uf} c^t_{b,s,f}, \forall t \in \mathcal{T}, u \in \mathcal{L}^b_u, b \in \mathcal{L}_b, f \in \mathcal{F}^t. \tag{6}$$

Per time slot $t,\ t \in \mathcal{T}$, the content demand f of user u is either dropped or satisfied from one of the locations with one of the net gains of $\{G^{u,t}_{b,f}, G^{u,t}_{b,b',f}, G^{u,t}_{b,s,f}\}$. Recall that the gross gains are inversely proportional to the latency of content fetching. By assigning different net gains with no units, i.e., $G^{u,t}_{b,f}$, $G^{u,t}_{b,b',f}$, or $G^{u,t}_{b,s,f}$, as the weighting factors to the transmission data rate, the backhaul traffic offloading, the cache hits as well as the content retrieving and content delivery can be jointly considered. Let us define the instantaneous serving reward, i.e., the weighted data rate, of SBS b for serving user u with content f at time slot $t,\ t \in \mathcal{T}$, as

$$\mathcal{R}_b(d_{uf}^t) = \begin{cases} G_{b,f}^{u,t} R_{bu}^t, & \text{if } c_{b,f}^t = 1, \\ G_{b,b',f}^{u,t} R_{bu}^t, & \text{if } c_{b,b',f}^t = 1, \\ G_{b,s,f}^{u,t} R_{bu}^t, & \text{if } c_{b,s,f}^t = 1, \\ 0, & \text{if } c_{b,s,f}^t + \sum_{b' \in \mathcal{L}_b} c_{b,b',f}^t = 0, \end{cases} \tag{7}$$

$$\forall b' \neq b, b \in \mathcal{L}_b, b' \in \mathcal{L}_b, u \in \mathcal{L}_u^b, f \in \mathcal{F}^t, t \in \mathcal{T}.$$

This serving reward can be regarded as the equivalent or effective data rate and will be useful in designing cache placement policy as well as content delivery policy in the subsequent sections.

III. PROBLEM FORMULATION AND DECOMPOSITION

A. Problem Formulation

Let us denote by $\mathbf{w}^t = \{\mathbf{c}_p^t, \mathbf{c}_r^t, \mathbf{c}_u^t\}$ the joint content caching, retrieving and delivery policy of the SBSs at time slot $t,\ t \in \mathcal{T}$. The objective of the CU is to design this policy $\{\mathbf{w}^t\}$ with joint consideration of backhaul traffic offloading, cache hit ratio, as well as content retrieving and delivery in the presence of the non-stationary content library. Hence, the problem of interest can be formulated as the maximization of the long-term average reward of the network, i.e., the average weighted network utility, as

$$\max_{\{\mathbf{w}^t\}} \left\{ \frac{1}{T} \sum_{t\in\mathcal{T}} \sum_{b\in\mathcal{L}_b} \sum_{f\in\mathcal{F}^t} \sum_{u\in\mathcal{L}_u^b} \ \mathcal{R}_b(d_{uf}^t) \right\} \tag{8}$$

$$\begin{aligned}
\text{s.t.} \quad & \text{C1}: \sum_{f\in\mathcal{F}^t} S_f c_{b,f}^t \leq M_b, \ \forall b \in \mathcal{L}_b, t \in \mathcal{T}, \\
& \text{C2}: c_{b,s,f}^t + \sum_{b'\in\mathcal{L}_b} c_{b,b',f}^t \leq 1, \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t, t \in \mathcal{T}, \\
& \text{C3}: c_{b,b',f}^t \leq c_{b',f}^t, \forall b \neq b', b \in \mathcal{L}_b, b' \in \mathcal{L}_b, f \in \mathcal{F}^t, t \in \mathcal{T}, \\
& \text{C4}: c_{b,s,f}^t \in \{0,1\}, \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t, t \in \mathcal{T}. \\
& \text{C5}: c_{b,b',f}^t \in \{0,1\}, \forall b \neq b', b \in \mathcal{L}_b, b' \in \mathcal{L}_b, f \in \mathcal{F}^t, t \in \mathcal{T}. \\
& \text{C6}: c_{b,u,f}^t \in \{0,1\}, \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t, u \in \mathcal{L}_u^b, t \in \mathcal{T}, \\
& \text{C7}: c_{b,f}^t \in \{0,1\}, \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t, t \in \mathcal{T}.
\end{aligned}$$

where the constraint C1 guarantees that the total size of the cached contents cannot exceed the capacity of cache units at the individual SBSs. C2 indicates that the content demands of the users will either be dropped or be satisfied from one of the SBSs' caches or from the content server. C3 denotes that SBS b can only retrieve content f from SBS b' if b' caches the requested content. C4 - C7 specify that the joint content caching, retrieving and delivery policy $\mathbf{w}^t$ is a binary vector.

1) Problem Analysis: The cross-time scale optimization problem in (8) is difficult to solve directly since we aim to maximize the long-term weighted network utility while the statistics of the system dynamics are unknown in advance. In general, the difficulties raised by the considered scenario are: the spatio-temporal unknown dynamics in users' content demands and channel conditions; the limited knowledge of new changes in the environment, e.g. limited samples of users' content requests, and the constrained caching space at the SBSs. In other words, at different time scales, the CU has to make decisions on which contents to cache as well as when and where to cache them, based on limited information of new changes in the presence of non-stationary environment. Hence, we are motivated to use RL technique to cope with these spatio-temporal uncertainties as it aims to maximize the cumulative reward via continually interacting with the environment and making sequential decisions of actions based on the reward (and state) information through the trial-and-error procedure.

2) Explanation of Non-Stationary MAB: The MAB problem that is regarded as a stateless RL problem [33], models a system of multiple arms (content library), each is associated with an unknown and stationary reward distribution. The agent (CU, on behalf of SBSs) makes sequential decisions on which contents to cache and aims to maximize the accumulated reward over time via exploring the environment by caching not frequently cached but potentially popular contents, while exploiting the current knowledge by caching contents associated with the highest rewards so far [33]. Here we consider a non-stationary variant of the MAB problem for our considered scenario with non-stationary content library, where the reward distributions of arms may vary across time.

B. Problem Decomposition and the Constrained Cross-Entropy Method

In the sequel, the cross-time scale optimization problem in (8) will be decomposed into RL-assisted optimization problems, and the joint cache placement, content retrieving and delivery policy $\{\mathbf{w}^t\}$ will be gradually optimized at different time scales through the proposed three-phase procedure in Section IV. More specifically, the content retrieving and delivery policy, i.e., $\{\mathbf{c}_r^t\}$ in constraints C2 - C5 of problem (8), will be satisfied at each individual time slot $t,\ t \in \mathcal{T}$, in Phase I of our proposed caching algorithm. The learning processes in Phase II and Phase III of the proposed algorithm, on the other hand, aim at tracking as much as possible the variations in user demands in order to design $\{\mathbf{c}_u^t\}$ in constraint C6 at every $\tau^{[\text{flex}]}$ time slots, and design $\{\mathbf{c}_p^t\}$ at every $\tau^{[\text{main}]}$ time slots, respectively.

Next, let us focus on Phase III for cache placement policy design, i.e., $\{\mathbf{c}_p^t\}$, at every $\tau^{[\text{main}]}$ time slots. As per (7), it is obvious that in order to maximize the long-term average reward of the network, the user will be served from the local cache of its serving SBS with the top priority, and by retrieving from the content server with the least priority. Hence, we can rewrite $c_{b,s,f}^t \leq 1 - \max_{b \in \mathcal{L}_b} c_{b,f}^t$, $c_{b,b',f}^t = c_{b',f}^t(1 - c_{b,f}^t)$, and the objective function of problem (8) as the expected overall reward among all SBSs, as

$$\mathcal{S}(\mathbf{c}_p^t) = \mathbb{E}\Bigg[\sum_{b \in \mathcal{L}_b} \sum_{f \in \mathcal{F}^t} \sum_{u \in \mathcal{L}_u^b} \mathcal{R}_b(d_{uf}^t)\Bigg] \leq \sum_{b \in \mathcal{L}_b} \sum_{f \in \mathcal{F}^t} S_f \upsilon_{b,f}^t \Bigg[(\pi^{[\text{local}]} - \kappa^{[\text{user}]} \sum_{u' \in \mathcal{L}_u^b, u' \neq u} c_{b,u',f}^t) c_{b,f}^t$$
$$+ \pi^{[\text{server}]}(1 - \max_{b \in \mathcal{L}_b} c_{b,f}^t) + \max_{b' \in \mathcal{L}_b, b' \neq b} \Bigg((\pi^{[\text{SBS}]} - \kappa^{[\text{user}]} \sum_{u' \in \mathcal{L}_u^{b'}} c_{b',u',f}^t) c_{b',f}^t (1 - c_{b,f}^t) \Bigg) \Bigg], \tag{9}$$

where $\upsilon_{b,f}^t$ is the expected value of $\{d_{uf}^t R_{bu}^t\}_{u \in \mathcal{L}_u^b}$ for content f at SBS b.

Due to the fact that $\{\upsilon_{b,f}^t\}$ is unknown and involves temporal dynamics, the non-stationary bandit technique will be employed in the following section to progressively improve the estimation of this value in Phase III of our proposed strategy. Let us denote by $\bar{\boldsymbol{\Upsilon}}^t = \{\bar{\upsilon}_{b,f}^t, \ \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t\}$ the estimated joint reward distribution of the SBSs over the non-stationary library of contents, and denote by $\bar{\mathcal{S}}(\mathbf{c}_p^t)$ the corresponding estimated expected overall reward among all SBSs, where $\bar{\mathcal{S}}(\mathbf{c}_p^t)$ can be obtained by replacing the unknown actual value $\upsilon_{b,f}^t$ in the right hand side of (9) with the estimated value of $\bar{\upsilon}_{b,f}^t$. As will be introduced in Section IV, $\bar{\boldsymbol{\Upsilon}}^t$ will be estimated at every $\tau^{[\text{main}]}$ time slots in Phase III of our proposed strategy, based on the past observations of the content demands and the transmission data rates. Then, with the estimated (learned) value of $\bar{\boldsymbol{\Upsilon}}^t$, the main content caching policy, i.e., $\{\mathbf{c}_p^t\}$, will be designed via the following content placement optimization problem, as

$$\begin{aligned} & \max_{\mathbf{c}_p^t} \ \bar{\mathcal{S}}(\mathbf{c}_p^t) && (10) \\ \text{s.t.} \quad & \text{C1}: \sum_{f \in \mathcal{F}^t} S_f c_{b,f}^t \leq M_b, \ \forall b \in \mathcal{L}_b, \\ & \text{C2}: c_{b,f}^t \in \{0,1\}, \ \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t. \end{aligned}$$

The problem in (10) can be regarded as a 0-1 knapsack problem with weights of $\{S_f\}_{f \in \mathcal{F}^t}$.

The 0-1 knapsack problem is a well-known NP-complete combinatorial optimization problem and the constraints satisfy monotonic property. Solving problem in (10) via either the branching algorithms such as the branch and bound (B&B) algorithm, or the semidefinite relaxation (SDR) approach [3] will generally require high computational complexity. Hence, we propose to solve problem in (10) with a near-optimal solution $\mathbf{c}_p^{t*}$ via the low-complexity CCE method. The cross-entropy (CE) method solves the maximization problem to the optimal or near-optimal solution by alternating between generating samples of random data according to a specified mechanism, and updating the parameters of the random mechanism based on the data in order to produce better samples in the next iteration [34]. However, the original CE method for unconstrained optimization cannot be applied directly to (10) in the presence of constraints, as many sample points might not be in the feasible region. Given the monotonic property of the constraints, we adopt the CCE method with the penalty approach[1] to solve the constrained problem in (10). The penalty approach relaxes the constraints in (10) in a similar fashion of the Lagrangian relaxation and artificially penalizes the evaluation of infeasible solutions via modifying the objective function in (10) as follows:

$$z^* = \max_{\mathbf{c}_p^t \in \mathcal{C}} \ \tilde{\mathcal{S}}(\mathbf{c}_p^t) = \max_{\mathbf{c}_p^t} \ \left\{ \bar{\mathcal{S}}(\mathbf{c}_p^t) - \sum_{b \in \mathcal{L}_b} H_b \max(\sum_{f \in \mathcal{F}^t} S_f c_{b,f}^t - M_b, 0) \right\}, \tag{11}$$

where the penalty parameter $H_b \gg 0$ indicates the importance of the penalty function and $\mathcal{C} \subset \mathbb{B}^{N_b F'}$ denotes the feasible region. The CCE method associates a stochastic estimation problem, i.e.,

$$\mathbb{P}(\tilde{\mathcal{S}}(\mathbf{c}_p^t) \geq z) = \sum_{\mathbf{c}_p^t \in \mathcal{C}} I_{\{\tilde{\mathcal{S}}(\mathbf{c}_p^t) \geq z\}} f(\mathbf{c}_p^t, \mathbf{p}), \tag{12}$$

where z is the worst value of $\tilde{\mathcal{S}}(\mathbf{c}_p^t)$ among N^{elite} elite (good-performing) samples in the previous iteration and is used as a threshold in the current iteration in order to generate better samples,

[1] By empirically adjusting the penalty parameter, the samples associated with infeasible solutions will be discarded accordingly in each iteration. The elite samples that violate the constraints can simply be projected onto the feasible region [35].

and $f(\mathbf{c}_p^t, \mathbf{p})$ is a Bernoulli distribution characterized by a parameter vector $\mathbf{p}$, as

$$f(\mathbf{x}, \mathbf{p}) = \prod_{j=1}^{n} (p_j)^{x_j} (1 - p_j)^{1-x_j}, \; x_j \in \{0, 1\}, j = 1, \cdots, n. \tag{13}$$

Hence, with increasing threshold value of z and via importance sampling, the estimated $\mathbb{P}(\tilde{\mathcal{S}}(\mathbf{c}_p^t) \geq z)$ converges either to the global optimum z^* or a value close to it. The steps and the computational complexity of each step of the CCE method are detailed in Algorithm 1. To be specific,

Algorithm 1 *CCE method for solving problem in* (10) *given the estimated* $\bar{\mathbf{\Upsilon}}^t$ [35]

```
1: Initialize: Stopping criteria $\delta$, iteration index $n = 1$, number of random samples $N_s$ ($N_s < N_b F'$), number of elite samples $N^{\text{elite}}$ ($N^{\text{elite}} \ll N_s$, typically 5%-10%), smoothing parameter $\alpha$ ($0.4 \leq \alpha \leq 0.9$), initial probabilities $\mathbf{p}^{[0]} = \{p_j^{[0]}\}_j \in (0, 1)$. ←-- $O(N_b F')$
2: REPEAT
3:     Sample: Generate $N_s$ random samples $\{\mathbf{x}_1, \cdots \mathbf{x}_j, \cdots \mathbf{x}_{N_s}\}$ from probability density function $f(\cdot, \mathbf{p}^{[n-1]})$. ←-- $O(N_b F' N_s)$
4:     Penalty Approach: Modify the objective function in (10) as per (11).
5:     Select: Sort samples in descending order with respect to values of $\tilde{\mathcal{S}}(\mathbf{c}_p^t)$. ←-- $O(N_s \text{log} N_s)$
       Select $N^{\text{elite}}$ elite (best-performing) samples that yield the top greatest values of $\tilde{\mathcal{S}}(\mathbf{c}_p^t)$.
6:     Update: For $j = 1 : N_b F'$, compute $\mathbf{p}^{[n]}$ as follows ←-- $O(N^{\text{elite}})$
         $p_j^{[n]} = \frac{\sum_{j=1}^{N_s} I_{\{\tilde{\mathcal{S}}(\mathbf{x}_j) \geq z\}} x_{ij}}{\sum_{j=1}^{N_s} I_{\{\tilde{\mathcal{S}}(\mathbf{x}_j) \geq z\}}} = \sum_{i \in \mathcal{I}} x_{ij} / N^{\text{elite}}$, where $\mathcal{I}$ is the index of $N^{\text{elite}}$ elite samples.
7:     Smooth: Update parameter vector $\mathbf{p}^{[n]}$, as $\mathbf{p}^{[n]} = \alpha \mathbf{p}^{[n]} + (1 - \alpha) \, \mathbf{p}^{[n-1]}$. ←-- $O(N^{\text{elite}})$
8:     Update $n = n + 1$.
9: UNTIL $\max_{f \in \mathcal{F}^t} (|\mathbf{p}^{[n]} - \mathbf{p}^{[n-1]}|) < \delta$
10: Output: Optimal main caching policy $\mathbf{c}_p^{t*} = \mathbf{p}^{[n]}$.
```

at each iteration n, the new value of z obtained from iteration $n - 1$ is used to update $\mathbf{p}^{[n]}$, whilst the updated vector $\mathbf{p}^{[n]}$ in turn, is used for generating better samples in iteration $n + 1$ as per steps 6 and 3 of Algorithm 1, respectively. The application of the smoothing parameter

α in step 7 is to prevent the occurrences of all zeros or all ones sub-optimal solutions, and the convergence in step 9 of Algorithm 1 can be achieved at a polynomial speed [35].

IV. The Proposed Foresighted Caching Strategy

In this section, the proposed RL-based user-assisted caching strategy is introduced to take joint consideration of the backhaul traffic offloading, the cache hits, as well as the content retrieving and content delivery, with the aims of keeping track of the dynamic content library and maximizing the long-term average reward as much as possible. Recall that the considered scenario raises challenges of the spatio-temporal unknown dynamics in user demands and channel conditions; the limited knowledge of new changes in the environment, and the constrained caching space at the SBSs. These difficulties involving temporal dynamics are handled in the following way. First of all, the caching problem is modelled as a non-stationary bandit problem, where the CU (on behalf of the SBSs) can be regarded as the agent, F' arms correspond to the current library of F' contents at the content server, and the associated reward of playing (requesting) the f-th arm can be defined as the aggregated content delivery rate for satisfying users' content demand of f. The standard upper confidence bound (UCB) algorithm [33] is modified to emphasize more on the recent observations. Secondly, we propose that SBSs' flexible cache memory can be updated by implementing a trade-off between caching new content from user cache directly (exploration), and updating flexible cache based on the knowledge of recent content demands (exploitation). Finally, content caching coordination among SBSs is enabled and the caching policy of SBSs is jointly designed at the CU to take full advantage of the capacity-constrained SBS cache units. The details of the proposed caching strategy are described in Algorithm 2 and Fig. 2, where a three-phase procedure at different time scales for joint content caching and delivery is proposed and explained below:

1) Phase I: At each time slot t, $t \in \mathcal{T}$, the users request contents from their neighboring SBSs. The SBS with the highest serving reward will be chosen as the serving SBS and the instantaneous

Algorithm 2 *User-assisted foresighted caching algorithm*

```
Initialize: $T$ time slots, temporary reward matrix $\mathbf{r}_b = \{r_{b,f}^t, \forall f \in \mathcal{F}^t, t \in \mathcal{T}\} = \mathbf{0}$, estimated mean reward $\bar{\mathbf{r}}_b^t = \{\bar{r}_{b,f}^t, \forall f \in \mathcal{F}^t\} = \mathbf{0}$, $\bar{\mathbf{\Upsilon}}^t = \{\bar{\upsilon}_{b,f}^t, \forall b \in \mathcal{L}_b, f \in \mathcal{F}^t\} = \mathbf{0}$, exploration/exploiation trade-off $\epsilon^t$, global/local trade-off $\mu^t$, discount factor $\beta$, weighting factor $\rho_b^{t=1} = 1$.
For $t = 1 : T$
If $t = \tau^{[\text{main}]}$, Phase III. Main Cache Placement
 CU updates $\bar{\mathbf{\Upsilon}}^t$ as $\bar{\upsilon}_{b,f}^t = \bar{r}_{b,f}^t + \rho_b^t \sqrt{\frac{2\log\ n_t}{T_{b,f}}}$, where $\rho_b^t \propto \sum_{t'=1}^{t} \sum_{u \in \mathcal{L}_u^b} \beta^{(t-t')} S_f d_{uf}^{[t']}$, $T_{b,f} = \sum_{t'=1}^{t} \beta^{(t-t')} I_{\{c_{b,f}^{[t']}=1\}}$ is the discounted number of times content $f$ has been cached so far and $n_t = \sum_{f \in \mathcal{F}^t} T_{b,f}$.
 Design cache placement policy via Algorithm 1 based on $\bar{\mathbf{\Upsilon}}^t$ and update SBSs' cache units in a sorted order.
End If
If $t = \tau^{[\text{flex}]}$, Phase II. Flexible Cache Update
 If A new content $f'$ is cached by the local user $u$ within the coverage area of SBS $b$
-with probability $\epsilon^t$, update SBS $b$'s flexible cache with content $f'$ directly;
-with probability $1 - \epsilon^t$, update $\bar{\mathbf{\Phi}}_b$ as per (14), and replace the flexible cache only if $\mathbf{\Phi}_b'(f')$ is larger than that of the contents in the flexible cache of SBS $b$.
 End if
End If
Phase I. Content Delivery at Each Time Slot $t$
 Users request contents $\{\mathbf{d}_u^t\}_{u \in \mathcal{L}_u}$ from their neighboring SBSs.
 If no SBS caches the requested content $f$
 -The SBS with the highest data rate serves the user by fetching content $f$ from the content server.
 Else If the requested content $f$ is cached at the SBS associated with the highest data rate
 -The SBS serves the user directly from its local cache.
 Else
 -The SBS associated with the highest weighted data rate (serving reward) serves the user.
 End if
 Update $\{\mathbf{r}_b\}$ as $r_{b,f}^t = \sum_{u \in \mathcal{L}_u^b} R_{bu}^t d_{uf}^t$, $\forall f \in \mathcal{F}^t, b \in \mathcal{L}_b$.
 Update $\{\bar{\mathbf{r}}_b^t\}$ as $\bar{r}_{b,f}^t = \frac{\sum_{t'=1}^{t} r_{b,f}^{t'} \beta^{(t-t')}}{\sum_{t'=1}^{t} \beta^{(t-t')}}, \forall f \in \mathcal{F}^t, b \in \mathcal{L}_b$.
End For
```

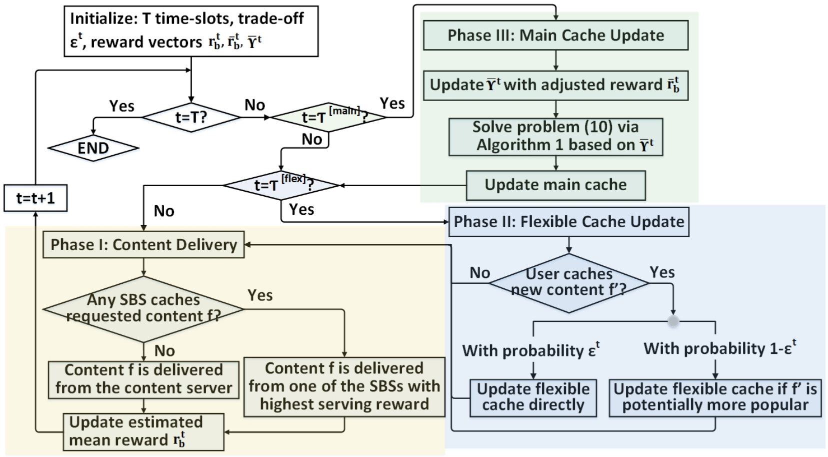

Initialize: T time-slots, trade-off ϵ^t, reward vectors r_b^t, $\bar{r}_b^t$, $\bar{Y}^t$
Yes
t=T?
No
$t=T^{[main]}$?
Yes
END
No
t=t+1
$t=T^{[flex]}$?
No
Yes
Phase III: Main Cache Update
Update $\bar{Y}^t$ with adjusted reward $\bar{r}_b^t$
Solve problem (10) via Algorithm 1 based on $\bar{Y}^t$
Update main cache
Phase II: Flexible Cache Update
No
User caches new content f'?
Yes
With probability ϵ^t
With probability $1-\epsilon^t$
Update flexible cache directly
Update flexible cache if f' is potentially more popular
Phase I: Content Delivery
Any SBS caches requested content f?
Yes
No
Content f is delivered from the content server
Content f is delivered from one of the SBSs with highest serving reward
Update estimated mean reward r_b^t

Fig. 2. Flowchart of the proposed RL-based user-assisted caching algorithm.

content demands are satisfied according to the delivery policy in step 15 of Algorithm 2. Then, the estimated mean rewards of the individual requested contents are updated as per step 17 of Algorithm 2. To be specific, step 17 calculates the average discounted accumulated content delivery rate for satisfying users' content demands of f at SBS b, which jointly considers user demand of content f as well as channel quality and user scheduling at SBS b.

2) Phase II: For every $\tau^{[\text{flex}]}$ time slots, the contents in SBSs' flexible caches will be replaced by the potentially more popular new contents cached at the users. More specifically, with the probability of $1-\epsilon^t$, the flexible cache will be updated accordingly as per step 10 of Algorithm 2, based on the following content demand vector:

$$\bar{\mathbf{\Phi}}_b = (1-\mu^t)\mathbf{\Phi}_c + \mu^t\mathbf{\Phi}_b,\ \forall b \in \mathcal{L}_b, \tag{14}$$

where $\mathbf{\Phi}_b$ denotes the recent local content demand vector at SBS b and $\mathbf{\Phi}_c = \sum_{b\in\mathcal{L}_b}\mathbf{\Phi}_b$ is the network wide recent content demand vector accumulated at the CU. The global/local trade-off $\mu^t,\ 0 \leq \mu^t \leq 1$, is employed to capture the spatial diversity of the content demands, such that the content caching coordination among SBSs can be capitalized. With the probability of ϵ^t, we explore new contents cached by local users and update SBSs' flexible caches directly.

The exploration/exploiation trade-off ϵ^t is tunable with respect to the temporal evolution rate of contents. More specifically, with a larger value of $N^{[\text{new}]}$ and the limited knowledge of new changes, a larger value of ϵ^t will be adopted to cache (explore) the new contents that may yield a better accumulated reward.

3) Phase III: For every $\tau^{[\text{main}]}$ time slots, a perturbation procedure is applied to the estimated mean reward $\bar{\mathbf{r}}_b^t$ according to step 4 in Algorithm 2. Such adjustment implements a trade-off between exploring the contents that are not frequently cached and may yield a better accumulated reward in the future by artificially increasing their estimated mean reward, and exploiting the contents associated with the highest mean reward so far based on the past observations. Due to the fact that the content library is massive and evolving, the standard soft-max and UCB algorithms that are designed based on the assumption of stationary and unknown reward distribution of individual arms may not be able to catch up with such rapid variations [36]. Hence, we modified the UCB-1 algorithm by adding a discount factor β [36] as well as a weighting factor ρ_b^t that is proportional to the long-term discounted content demands. Such modification will encourage the SBSs to cache those contents that are frequently requested in recent times but are not cached that often. Specifically, a smaller value of β will be applied to emphasize more on the recent observations when content evolution rate increases.

A. Computational Complexity Analysis

The computational burden of the proposed algorithm mainly lies in optimizing the cache placement policy in step 5 of Algorithm 2 via Algorithm 1. As stated in the previous section, the optimal or near optimal solution of problem in (10) can be found via the B&B algorithm, the SDR approach [3] or the CCE method. The worst case complexity of the B&B algorithm is $O(F'^{\sum_{b\in\mathcal{L}_b} M_b})$, which is the same as that of the exhaustive search [26]. The SDR approach relaxes problem in (10) as a semidefinite programming problem, which can be solved via the interior-point algorithm with a worst-case computational complexity of $O(\max\{\tau_s, N_b(F'+1)\}^4\tau_s^{0.5}log(\frac{1}{\sigma}))$ [3], where σ is the solution accuracy and τ_s denotes the problem size of (10).

In addition, a recovery approach is necessary to recover the rank-one solution and to reconstruct the optimal caching decisions $\mathbf{c}_p^t$, which will further increase the computational complexity of the problem. The overall computational complexity of the CCE method in Algorithm 1 is $O(N_b F' M_b N_s \log N_s)$, and the main complexity lies in the performance evaluation of N_s samples for the modified objective function $\tilde{\mathcal{S}}(\mathbf{c}_p^t)$ as per step 5 of Algorithm 1. Thus, the complexity is low and can be further reduced through a trade-off between the complexity and solution accuracy.

B. Signalling Overhead Analysis

Recall that the CU coordinates all content caching and delivery strategies based on channel gain and user content demand information uploaded at each individual time slot from the SBSs. In general, the signalling overhead of the proposed design consists of the following information exchanges between the CU and the SBSs at each time slot t: (1) channel gain information $\{\Psi_{bu}^t\}$ uploaded from the SBSs to the CU; (2) user content demand information uploaded from the SBSs to the CU; and (3) decisions on serving SBSs for the individual users dispatched from the CU to the SBSs. The average signalling overheads incurred by the above information exchanges at each time slot are, respectively, $O(N_b K)$, $O(K)$ and $O(K)$. In addition, the users' cached content directory is updated from the SBSs to the CU at every $\tau^{[\text{flex}]}$ time slots, and the CU will then send control commands on flexible cache update to the corresponding SBSs. The resulting signalling overheads are, respectively, $O(K)$ and $O(2\xi M_b N_b)$ at the most.

C. Performance Discussion and Future Directions

The dynamic regret analysis [31] of the standard discounted UCB algorithm has been conducted in [36], where the upper-bound of the expected regret is established by upper-bounding the expected number of times the suboptimal arms are selected. However, the regret analysis is more challenging for our considered scenario, due to the fact that we consider multiple time scales for main cache update from the content server, flexible cache update from the user caches, and content delivery. Unlike standard MAB problem where one arm is played at each time, we cache multiple (differentiated) contents at multiple SBSs via content caching coordination among SBSs during Phase III and replace some of the cached content with user cache during Phase II. Hence, the regret analysis and/or performance guarantee is beyond what can be covered in this paper, and we would like to leave it for our future work.

V. SIMULATION RESULTS

Consider a downlink small cell network comprising 3 neighbouring SBSs that serve $K = 12$ randomly deployed users. The non-stationary content library has an initial library of $F = 200$ contents and a finite capacity of $F^{[\text{max}]} = 250$. The Phase III for SBSs' main cache update occurs every $\tau^{[\text{main}]} = 8$ time slots, where the capacity of caching unit at each SBS is $M_b = 15$ with $\xi = 0.2$. The Phase II for flexible cache update takes place every $\tau^{[\text{flex}]} = 3$ time slots, where $N^{[\text{new}]} = 2$ new contents will be added to the content server and might be cached by at most 2 random users. The per-unit gross gains for SBSs to serve users directly from their local caches, by fetching contents from caches of the other SBSs and by retrieving contents from the content server are, respectively, $\pi^{[\text{local}]} = 1$, $\pi^{[\text{SBS}]} = 0.5$ and $\pi^{[\text{server}]} = 0.1$, whilst the per-unit average discount rate for users' uploading incentives is $\kappa^{[\text{user}]} = 0.1$. The users' content request arrival processes are modeled as independent homogenous Poisson point processes with request rate of 1 [32]. We adopt a classical independent reference model, i.e., the commonly used power-law Zipf distribution [7], given by

$$\theta^{t}_{u\to\Delta_u,f} = \frac{f^{-\gamma^t}}{\sum_{f=1}^{F'} f^{-\gamma^t}}, \;\; f \in \mathcal{F}^t, u \in \mathcal{L}_u, \tag{15}$$

to model the actual content popularities of the users at time t that are unknown to the SBSs, where $\gamma^t = 2.5$ is the Zipf exponent indicating the popularity skewness. The shift of content popularity distribution at user u with respect to user $u-1$, i.e., Δ_u, is randomly drawn from $\{0, 1, 2\}$. Note that we employ the Zipf distribution in the simulation just as an illustration to evaluate our proposed caching algorithm, and the choice of the content popularity distribution model will not affect the effectiveness of our proposed algorithm. The channel gain is modelled as $\Psi_{bu}^{t} = \mathbf{h}_{bu}^{t2} G_a L_{bu} e^{-0.5\frac{(\sigma_s \ln 10)^2}{100}}$, where $\mathbf{h}_{bu}^{t} \sim \mathbb{CN}(0,1)$, $L_{bu}(\text{dB}) = 128.1 + 37.6\log_{10}(\ell)$ [37] is the path loss model over a distance of ℓ km between SBS b and user u, $G_a = 15$ dBi and $\sigma_s = 10$ dB denote, respectively, the antenna gain and the log-normal shadowing standard deviation. The other simulation parameters are described, unless otherwise stated, as follows: coverage radius $R_b = 500$ m, transmit power $P_b^{[\text{Tx}]} = 20$ dBm, $N = 100$ random samples, $N^{\text{elite}} = 10$ elite samples, smoothing parameter $\alpha = 0.9$, exploration/exploitation trade-off $\epsilon^t = 0.6$, discount factor $\beta = 0.93$ and global/local trade-off $\mu^t = 0.7$. The proposed strategy is evaluated with $T = 1000$ time slots for each set of parameter setting. Five designs that consider no user cache are chosen as the benchmark designs, namely, the algorithm in [3], the algorithm in [26], the EXP3-based caching design, the local popularity-based caching design and the random caching design. All benchmark designs follow similar procedures of main cache update (Phase III) and content delivery (Phase I) as for our proposed strategy, whereas, no user cache exploitation (Phase II) is considered in the benchmark designs. For fair comparison, identical constraints have been applied and the performance metrics, i.e., the average weighted network utility, are the same for all strategies.

1) Benchmark design in [3]: The content popularity distribution is estimated via the standard UCB-1 algorithm [33]. The estimated mean reward is set as the estimated popularity distribution, given by $\bar{\Theta}_{b,f}^{t} = \frac{O_{b,f}^{t}}{N_{b,f}^{t}}$, where $O_{b,f}^{t}$ and $N_{b,f}^{t}$ denote, respectively, the long-term observation of the request number of content f in SBS b, and the total number of time slots the requests of content f are satisfied by local SBSs' caches.

2) Benchmark design in [26]: The content popularity distribution is learned via the combinatorial UCB algorithm based on the past observation of user content demands. The estimated mean reward is given by $\bar{\Theta}_{b,f}^{t} = \frac{\sum_{t'=1}^{t}\sum_{u\in\mathcal{L}_u^b} S_f d_{uf}^{t'}}{T_{b,f}}$.

3) EXP3-based caching design: The Exponential-weight algorithm for Exploration and Exploitation (EXP3)-based caching design caches contents via softmax action selection policy [38].

More specifically, a list of weights are assigned to the individual contents and are adjusted based on the instantaneous reward $r^t_{b,f}$. These weights are then utilized in a softmax-weighted manner to decide randomly which contents to cache during the main cache update phase. For fair comparison, the above three benchmark designs are embedded with a sliding-window [36], which emphasizes more on the local empirical average of the recent observed rewards, so as to better adapt to our considered scenario.

4) Local popularity-based caching design: It estimates the content popularity distribution at the individual SBSs in a distributed way without any signalling with the CU, and caches contents merely based on recent local content demand observations.

5) Random caching design: It randomly caches contents at the individual SBSs without considering any content caching coordination among SBSs. This design is employed to indicate the lower bound and to demonstrate the advantage of the cooperative caching and joint optimization of different caching locations.

6) Optimal caching design: For better evaluation of the proposed caching strategy, we further adopt a user-aided optimal caching design to show the performance upper bound. The optimal caching design has perfect prior knowledge of the actual content popularity distributions $\{\theta^t_{u,f}\}_{f\in\mathcal{F}^t}$, and allows the SBSs to update their cache units in Phase III from the content server as well as their flexible caches in Phase II with local users' cached contents based on $\{\theta^t_{u,f}\}_{f\in\mathcal{F}^t}$.

Fig. 3 illustrates the comparison of average reward, i.e., the weighted network utility, of the proposed caching strategy against various benchmark designs at the individual time slots. As can be observed from the figure, the proposed strategy outperforms all of the five benchmark designs due to the negligence of both the evolution of the content library and the potentiality of user caches in the nature of their designs. To be specific, the benchmark designs in [3], [26] and the EXP3-based caching design, respectively, employ standard UCB and EXP3 algorithms, which are originally designed for stationary reward distributions, thus provide poorer adaptation to the non-stationary content library. Furthermore, they merely focus on designing the content placement

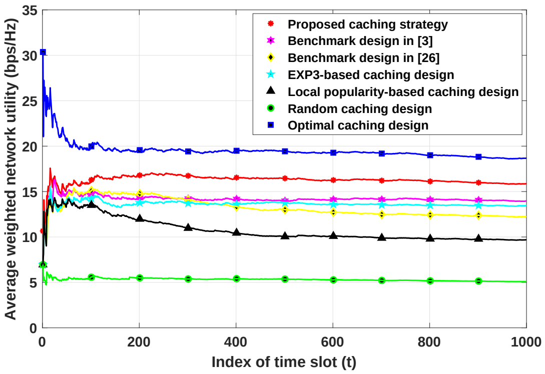


Fig. 3. Comparison of average weighted network utility for different strategies at individual time slots.

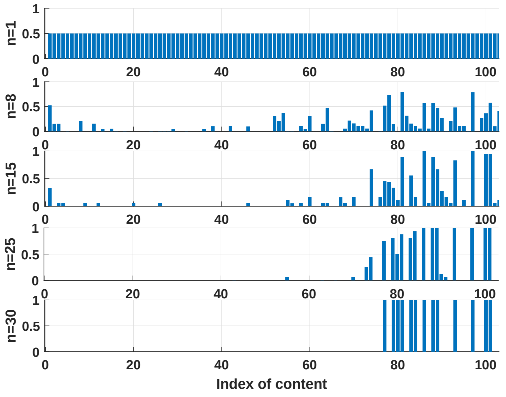
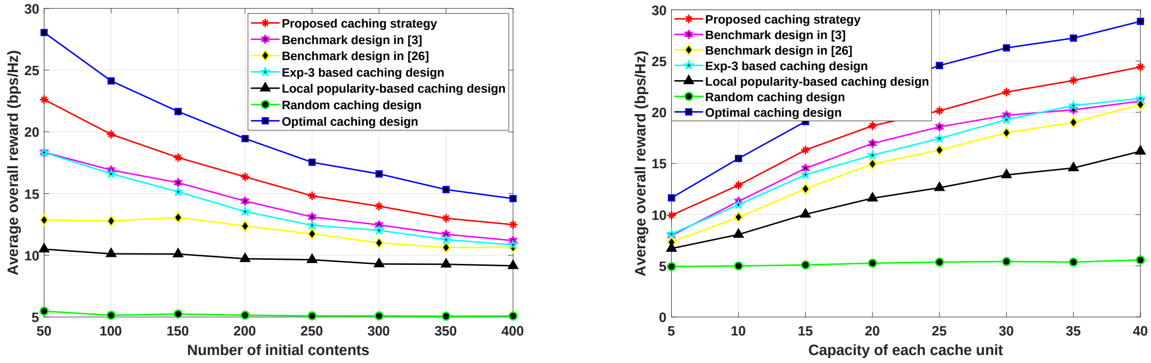


Fig. 4. Evolution of content placement policy via the CCE method.

policies, whilst ignoring the network utility for content delivery, thus have worse performance than our proposed design under the performance metric of average weighted network utility. Meanwhile, the local popularity based caching design and the random caching design have the worst performance among all designs. The reason is that they do not involve any signalling with the CU hence have no centralized content caching coordination among SBSs. The former caches contents merely based on recent local content demand observations, whilst the latter simply caches contents randomly at the SBSs without any learning process to estimate the unknown variations in user demands. In contrast, our proposed strategy, at the cost of light signalling overhead, takes into account the spatio-temporal variations in users' content demands, and maximally benefits from the user caches through timely updating the SBS's flexible cache in addition to the main cache update from the server, thus provides a better adaptation to the user demand variations.

Fig. 4 provides an illustration of the evolution of content placement policy of SBS 1 at the 1^{st}, the 8^{th}, the 15^{th}, the 25^{th} and the 30^{th} iterations of Algorithm 1 at the 9^{th} time slot. It is clear from the figure that by updating and smoothing the parameter vector $\mathbf{p}^{[n]}$ as per step 6 and step 7 in

Algorithm 1, respectively, better random samples can be produced in the subsequent iterations. Furthermore, the CCE method in Algorithm 1 converges within approximately 30 iterations and the outputs at the 25^{th} iteration are close to the converged solutions, which indicates a much lower complexity and a faster convergence speed as compared to the B&B algorithm.

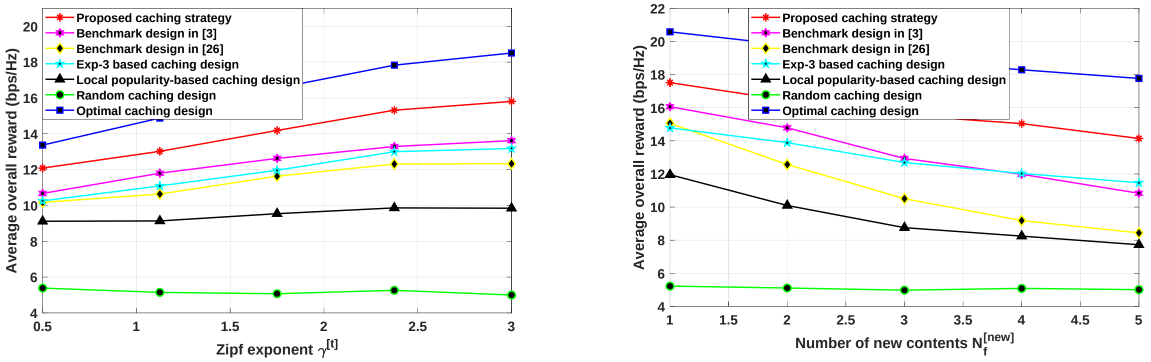


(a)

(b)

Fig. 5. Comparison of average overall reward for (a) various number of initial contents, (b) different capacity of individual cache units at the SBSs.

Fig. 5(a) and Fig. 5(b) respectively, compares the average overall reward, i.e., the overall weighted network utility averaged over $T = 1000$ time slots, of the proposed strategy against all benchmark designs for different initial sizes of content library and for various storage capacity at the SBSs. The initial size of library ranges from $F = 50$ to $F = 400$, with the finite content library capacity set to be $F^{[\max]} = F + 50$, and the storage capacity at the SBSs ranges from 5 to 40. As can be observed from Fig. 5, the proposed strategy has a better average overall reward as compared to the benchmark designs, since neither the time-varying content popularity nor the non-stationary content library has been taken into consideration in their designs. Furthermore, one may conclude from Fig. 5(a) that the average performance of all strategies degrades with the increasing number of initial contents, due to the fact that larger content library naturally results in more users' content requests being satisfied by the content server. On the other hand, the performance of all strategies improves with the increasing capacity of the individual SBSs' cache units in Fig. 5(b). The reason is that with larger caching space, the SBSs can cache more (differentiated) popular contents locally and reduce duplicate data transmission from the content server, and thus, offload more traffic from the content server to the edge.



(a)

(b)

Fig. 6. Comparison of average overall reward for different (a) content popularity variations, (b) number of new contents.

Fig. 6(a) and Fig. 6(b) present the average overall rewards of all strategies for various actual content popularity variations and different number of emerged new contents $N^{[\text{new}]}$, respectively. It is evident from Fig. 6(a) that both the proposed strategy and the learning based benchmark designs have improved performance with the increasing value of γ^t, whilst less influence is observed for the random caching design under different values of γ^t. More specifically, with larger value of γ^t and more diverse content popularities, the majority of the content demands of the users are occupied by fewer most frequently requested contents, hence is more favourable for the learning based caching designs. As can be concluded from Fig. 6(b), though having better performance as compared to the random caching design, the average overall rewards of the proposed strategy as well as other benchmark designs decrease with the increasing number of $N^{[\text{new}]}$. This is due to the fact that with a larger value of $N^{[\text{new}]}$, it is more challenging for the learning process to catch up with such rapid changes in users' content demands, especially when the knowledge of new changes is limited, e.g. limited samples of users' new content requests, and when the local caching space is constrained. However, the proposed strategy, as compared to the benchmark designs, is more robust in coping with the spatio-temporal variations of user demands. On the contrary, the variations in $N^{[\text{new}]}$ have less impacts on the random caching design as it fails to satisfy users' content requests for most of the time.

VI. CONCLUSION

The joint cache placement and content delivery problem in small cell networks is studied in this paper, where the spatio-temporal dynamic content popularity is unknown *a priori* and the content library evolves over time. To take the capacity-constrained cache units at the SBSs into account, content caching coordination among SBSs is adopted to improve the caching performance. To keep track of the dynamic content library, a portion of each cache unit is assigned as the flexible cache that can be timely updated with the contents cached by users in addition to the routine off-peak main cache update from the content server. Considering three phases of different time scales for the content delivery, the flexible cache update and the main cache update, the problem of interest is modelled as a RL-assisted optimization problem and a user-assisted caching algorithm is proposed to maximize the long-term average weighted utility of the network. Simulation results confirm the superiority of the proposed caching strategy in achieving a significant performance improvement over various benchmark designs.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2016-2021", White Paper, Sep. 2017.

[2] X. Zhang, G. Zheng, S. Lambotharan, M. R. Nakhai and K-K. Wong, "A Learning Approach to Edge Caching with Dynamic Content Library in Wireless Networks", *IEEE GLOBECOM*, Dec. 2019.

[3] J. Song, M. Sheng, T. Q. S. Quek, C. Xu and X. Wang, "Learning-Based Content Caching and Sharing for Wireless Networks," *IEEE Transactions on Communications*, vol. 65, no. 10, pp. 4309-4324, Oct. 2017.

[4] C. Fang, F. R. Yu, T. Huang, J. Liu and Y. Liu, "A Survey of Energy-Efficient Caching in Information-Centric Networking," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 122-129, Nov. 2014.

[5] L. Li, G. Zhao and R. S. Blum, “A Survey of Caching Techniques in Cellular Networks Research Issues and Challenges in Content Placement and Delivery Strategies,” *IEEE Comm. Surveys & Tutorials*, vol.20, no.3, pp.1710-1732, Mar.2018.

[6] E. Bastug, M. Bennis and M. Debbah, “Living on the Edge: The Role of Proactive Caching in 5G Wireless Networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82-89, Aug. 2014.

[7] Y. Zhu, G. Zheng, L. Wang, K-K. Wong and L. Zhao, “Content Placement in Cache-Enabled Sub-6 GHz and Millimeter-Wave Multi-Antenna Dense Small Cell Networks,” *IEEE Trans. on Wireless Com.*, vol.17, no.5, pp.2843-2856, Oct.2018.

[8] B. Blaszczyszyn and A. Giovanidis, “Optimal Geographic Caching in Cellular Networks,” *IEEE ICC*, Jun. 2015.

[9] J. Gu, W. Wang, A. Huang, H. Shan and Z. Zhang, “Distributed Cache Replacement for Caching-Enable Base Stations in Cellular Networks,” *IEEE International Conference on Communications (ICC)*, pp. 2648-2653, Jun. 2014.

[10] M. Afshang, H. S. Dhillon and P. H. J. Chong, “Fundamentals of Cluster-Centric Content Placement in Cache-Enabled Device-to-Device Networks,” *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2511-2526, Jun. 2016.

[11] R. Wang, X. Peng, J. Zhang and K. B. Letaief, “Mobility-Aware Caching for Content-Centric Wireless Networks: Modeling and Methodology,” *IEEE Communications Magazine*, vol. 54, no. 8, pp. 77-83, Aug. 2016.

[12] B. Zhou, Y. Cui and M. Tao, “Stochastic Content-Centric Multicast Scheduling for Cache-Enabled Heterogeneous Cellular Networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6284-6297, Sep. 2016.

[13] Y. Cui and D. Jiang, “Analysis and Optimization of Caching and Multicasting in Large-Scale Cache-Enabled Heterogeneous Wireless Networks,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 250-264, Jan. 2017.

[14] M. Tao, E. Chen, H. Zhou and W. Yu, “Content-Centric Sparse Multicast Beamforming for Cache-Enabled Cloud RAN,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6118-6131, Sep. 2016.

[15] S-H. Park, O. Simeone and S. S. Shitz, “Joint Optimization of Cloud and Edge Processing for Fog Radio Access Networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7621-7632, Nov. 2016.

[16] N. Karamchandani, U. Niesen, M. A. Maddah-Ali and S. N. Diggavi, “Hierarchical Coded Caching,” *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212-3229, Jun. 2016.

[17] Q. Yan, U. Parampalli, X. Tang and Q. Chen, “Online Coded Caching With Random Access,” *IEEE Communications Letters*, vol. 21, no. 3, pp. 552-555, Mar. 2017.

[18] R. Pedarsani, M. A. Maddah-Ali and U. Niesen, “Online Coded Caching ,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836-845, Apr. 2016.

[19] B. Chen, C. Yang and G. Wang, “High-Throughput Opportunistic Cooperative Device-to-Device Communications With Caching,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7527-7539, Aug. 2017.

[20] Z. Zhao, M. Peng, Z. Ding, W. Wang and H. Vincent Poor, “Cluster Content Caching: An Energy-Efficient Approach to Improve Quality of Service in Cloud Radio Access Networks,” *IEEE JSAC*, vol. 34, no. 5, pp. 1207-1221, May 2016.

[21] B. Chen, C. Yang and Z. Xiong, “Optimal Caching and Scheduling for Cache-Enabled D2D Communications,” *IEEE Communications Letters*, vol. 21, no. 5, pp. 1155-1158, May 2017.

[22] K. Poularakis, G. Iosifidis, V. Sourlas and L. Tassiulas, “Exploiting Caching and Multicast for 5G Wireless Networks,” *IEEE Transactions on Wireless Communications*, vol.15, no.4, pp.2995-3007, Apr. 2016.

[23] C. Yang, Y. Yao, Z. Chen and B. Xia, “Analysis on Cache-Enabled Wireless Heterogeneous Networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 131-145, Jan. 2016.

[24] E. Bastug, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S. Er and M. Debbah, “Big Data Meets Telcos: A Proactive Caching Perspective,” *Journal of Communications and Networks*, vol. 17, no. 6, pp. 549-557, Dec. 2015.

[25] J. Kwak, Y. Kim, L. B. Le and S. Chong, “Hybrid Content Caching in 5G Wireless Networks: Cloud Versus Edge Caching,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3030-3045, May 2018.

[26] P. Blasco and D. Gunduz, “Learning-based Optimization of Cache Content in a Small Cell Base Station,” *IEEE International Conference on Communications (ICC)*, pp. 1897-1903, Jun. 2014.

[27] A. Sadeghi and F. Sheikholeslami and G. B. Giannakis, “Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities,” *IEEE Journal of Selected Topics in Signal Proc.*, vol.12, no.1, pp.180-190, Feb.2018.

[28] B. N. Bharath, K. G. Nagananda, D. Gunduz and H. Vincent Poor, “A Learning-Based Approach to Caching in Heterogenous Small Cell Networks,” *IEEE Transactions on Communications*, vol. 64, no. 4, pp. 1674-1686, Apr. 2016.

[29] S. Tamoor-ul-Hassan, S. Samarakoon, M. Bennis, M. Latva-aho and C. S. Hong, “Learning-Based Caching in Cloud-Aided Wireless Networks,” *IEEE Communications Letters*, vol. 22, no. 1, pp. 137-140, Jan. 2018.

[30] M. Garetto, E. Leonardi, and S. Traverso, “Efficient Analysis of Caching Strategies under Dynamic Content Popularity,” *IEEE INFOCOM*, Apr. 2015, pp. 2263-2271.

[31] J. Yuan and A. Lamperski, “Trading-Off Static and Dynamic Regret in Online Least-Squares and Beyond," AAAI 2020.

[32] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi and S. Niccolini, “Temporal Locality in Today’s Content Caching: Why it Matters and How to Model it", *ACM SIGCOMM Computer Com. Review*, vol.43, no.5, pp.5-12, Oct.2013.

[33] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction," Cambridge MA, US: The MIT press, 2017.

[34] C. R. Rao and V. Govindaraju, “Chapter 3 - The Cross-Entropy Method for Optimization", *Machine Learning: Theory and Applications*, Elsevier, 2013.

[35] M. Caserta, E. Quinonez Rico and A. Marquez Uribe, “A Cross Entropy Algorithm for the Knapsack Problem with Setups", *Computers & Operations Research*, vol. 35, no. 1, pp. 241-252, Aug. 2008.

[36] A. Garivier and E. Moulines, “On Upper-Confidence Bound Policies for Non-Stationary Bandit Problems," *International Conference on Algorithmic Learning Theory*, pp. 174-188, 2011.

[37] 3GPP, “TR 36.814 V9.2.0: Further Advancements for E-UTRA Physical Layer Spects (Release 9)," Mar. 2017.

[38] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *European Conference on Computational Learning Theory*, pp. 23-37, 1995.