Caching Dynamic Contents via Mortal Restless Bandits

Anu Krishna
Department of Electronic Systems Engineering
Indian Institute of Science
Bangalore, India
vasanthakuma@iisc.ac.in

Chandramani Singh
Department of Electronic Systems Engineering
Indian Institute of Science
Bangalore, India
chandra@iisc.ac.in

***Abstract*—We study content caching in a network consisting of a server and a base station with a finite-capacity cache. Contents are dynamic. They stochastically arrive in the system, stay for random times, and their popularities also randomly vary while they are alive. Fetching contents from the server and storing in the base station cache incurs a cost. But not having the requested contents at the base station also incurs a cost that reflects QoS deficiency. We study optimal proactive caching to minimise the time-average content miss and caching costs. We formulate this problem as an average cost Markov decision problem that is a restless multi-armed bandit problem. We argue that the problem is indexable and explicitly derives the Whittle indices. Finally, we demonstrate the efficacy of the Whittle index policy via numerical evaluation.**

***Index Terms*—caching, restless bandits, threshold policy, whittle index, dynamic popularity**

I. Introduction

Over the last few years, there has been a rampant increase in demand for video-on-demand (VoD) services. According to a recent McKinsey and Company report, technology industry analysts predict further growth in VoD services and expect to see the market for Software as a service (SaaS) products to approach near 200 billion by 2024 [1]. In the context of VoD services, some videos may be more popular than others. These and other characteristics suggest that caching should improve the performance and scalability [2], [3]. There is plenty of literature that has studied caching of VoD services.

Adding a cache to the Base Stations (BSs) invariably offers several advantages. Caching contents at a BS closer to the users minimizes delay in fetching the requested content as it would be quicker to fetch them from the BS than from the central server. As far as the users are concerned, this can significantly improve their Quality of Experience (QoE). Also, caching reduces traffic over the backhaul links that connect BSs to the servers.

Despite the advantages offered by caching, there is a trade-off. Caching all the contents from the server at the BSs is not viable because the caches have a finite capacity. Also, caching contents from the server incurs a certain caching cost. Therefore caching contents that are less popular or would seldom be requested is undesirable. In general, in all caching problems, this is the motivation to develop an appropriate caching strategy to determine which contents should be cached.

Two types of caching strategies have been proposed in the literature, reactive and proactive caching strategies. In reactive caching, contents are cached only on request. In a proactive caching strategy, contents can be proactively cached even before being requested based on their popularity and dynamics of popularity. Proactive caching is preferred to reactive caching in order to minimize cache miss. However, most of the work related to proactive caching has assumed that the contents' popularity is fixed. On the other hand, we let content popularity evolve with time.

We consider a scenario where contents arrive at the server according to a stochastic process and their popularity evolution and extinction follow a discrete-time Markov chain (DTMC). We assume fixed content fetching cost and per request content miss cost for each content. We frame the optimal proactive caching problem as an average cost Markov decision problem (MDP) that is a restless multi-armed bandit (RMAB) problem [4]. However, solving a restless multi-armed bandit problem is, in general, difficult as it suffers from the curse of dimensionality [5]. Therefore, we employ a low-complexity heuristic called the Whittle index policy. The Whittle index policy has been extensively used in literature, e.g., see [6]–[8]. Furthermore, it is known to have a strong performance in a range of applications [9]–[11]. We propose to use the Whittle index policy for solving the problem of optimal caching.

In our RMAB problem, the MDP associated with a single arm is closely related to the MDP in [12]. The key difference between the two problems is that while [12] focuses on a discounted cost MDP, we have a stochastic shortest path problem at hand with content extinction being the terminal state.

A. Related Work

1) Content Caching: Most of the literature on content caching in networks has focused on the problem of delivering multimedia content such as music or videos by deploying caches close to the end users [3]. Such contents have time-varying popularity and it affects edge caching decisions [13]. Constant popularity drawn from Zipf distribution is hardly useful even in the context of a fixed set of contents with time-varying popularity [14]. Therefore, we cannot employ Zipf distribution in our content caching problem where the number of contents at the server evolves over time. In [15], the authors prove that chunking of videos is helpful. They

ISBN 978-3-903176-55-3 © 2023 IFIP

propose an algorithm called chunk-LRU for caching partial files and use an independent reference model (IRM) to model the content request process. However, it is shown in [16] that there are better models than an IRM to characterize the dynamic popularity of content.

While there is no dearth of literature on caching dynamic content, the term "dynamic" can have different meanings depending on the context in which it is used. For instance, the paper by [17]–[19] addresses the problem of content caching in dynamic environments where the popularity of content is unknown and evolves over time. The authors propose Reinforcement Learning based caching strategies. On the other hand, in [20]–[22], the contents are dynamic in the sense that the contents at the server receive updates. The server always retains a fresh version of all contents. The aforementioned dynamic caching problems assume that the server hosts a huge but fixed number of contents.

Several other approaches have been proposed, including online caching based on the history of request arrivals [23], federated learning-based caching [24], dynamic probabilistic caching policies [14], caching models accommodating time-varying request rates [25], online caching algorithms based on content request predictions [26], and joint optimization of service caching and routing [27]. Meanwhile, [12] and [28] accounted for temporal changes in content popularity by modelling the content request process as a stochastic process whose intensity is modulated by an underlying discrete-time Markov chain. The critical difference between our work and [12], [14], [23]–[28] is our assumption that new contents stochastically arrive and the existing ones expire after random lifetimes.

In a more realistic setting, the number of contents at the server evolves over time [29]. In our problem, contents are dynamic in the sense that the contents arrive at the server according to a stochastic process and their popularity evolution and extinction follows a discrete-time Markov chain. Our previous work [30] studies a caching problem where the number of contents evolves over time. However, in [30], the contents' instantaneous request rates are assumed to be a function of the number of contents at the server. Consequently, at any given time, all the contents have the same popularity. In our current work, the request rates of the contents vary stochastically and are independent of the number of contents at the server. The request rates of different contents are also independent of each other. We formulate this problem as a restless multi-armed bandit (RMAB) problem.

2) Restless Multi-armed Bandits: In an RMAB, a decision maker determines which K arms out of the total N arms should be activated at any time. The decision maker knows the states of all bandits and the cost in every state and aims to minimize the discounted or time-average cost over an infinite time horizon. The state of a bandit evolves stochastically according to transition probabilities that depend on whether the bandit is active or not. In principle, an RMAB problem can be solved by dynamic programming, but this approach is not tractable for realistic model sizes because of computational complexity [5]. Whittle [4] developed a methodology to obtain a heuristic by solving a relaxed version of the RMAB in which K arms are activated only on average. This heuristic, known as the Whittle index policy, relies on calculating the Whittle index for each of the bandits and activating at every decision epoch K arms with the highest Whittle indices. However, the Whittle indices are defined only when a certain indexability condition is satisfied which, in general, is hard to verify.

Whittle [31] and Weiss [32] also introduced open bandit processes where new arms continually arrive. Later the authors in [33] considered mortal multi-armed bandits whose arms have (stochastic) lifetime after which they expire. However, unlike our work, the bandits in these works are not restless. While the authors in [34] formulate a content caching problem as a RAMB problem to minimize the average content service latency, they assume a fixed number of contents at the server.

B. Contributions

We take a novel approach to the caching problem where the contents randomly arrive at the server and stay there until it expires. We propose dynamic caching policies where if the requested content is not available at the BS, it is fetched from the server and stored in the cache at the BS, keeping in view of the capacity of the cache. We cast our problem as Mortal MAB which is a variant of RMAB. We use a low complexity policy called the Whittle Index policy to solve our problem since RMAB is intractable. However, the Whittle Index policy is defined only for problems that are indexable, which is often challenging to establish.

We establish that the relaxed RMAB is indexable and obtain a closed-form expression for Whittle's Index. This allows us to develop a Whittle index policy for the caching problem. In addition, we compare the performance of the Whittle index policy with greedy policy and policy under relaxed RMAB.

II. System Model and Caching Problem

We consider a wireless network where the users are connected to a single base station (BS) which in turn is connected to a content server via the core network. We assume a slotted system. In particular, caching decisions are taken at the slot boundaries. Below we present content dynamics and the request model.

A. System Model

Content dynamics: New contents randomly arrive at the content server and leave after random lifetimes. During their lifetimes, i.e., before being extinct, the contents can be in low or high-popularity states, denoted as 1 and 2, respectively. While a more granular level of popularity might better reflect real-world scenarios, to simplify the problem, we have restricted the number of popularity levels to two, as suggested in [35] and [12].

We assume that the contents arrive at the server according to an *independent and identically distributed* (i.i.d.) arrival sequence $m(t) = (m_1(t), m_2(t)), t \geq 0$. Let $m_s(t)$, $s \in \{1, 2\}$, be the number of contents that arrive during $[t-1, t]$ and are in state s in this slot, with $\mathbb{E}[m_s(t)] = \lambda_s(t)$, $s \in \{1, 2\}$. A content's popularity evolution and extinction constitute a discrete-time Markov chain (DTMC), as shown in Figure 1;

ISBN 978-3-903176-55-3 © 2023 IFIP

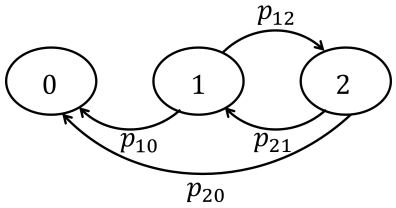


Fig. 1: Popularity evolution of a content. 1 and 2 denote low and high popularity states, respectively. 0 denotes the terminal *extinction* state. Self-loops are omitted for clarity.

here 0 denotes extinction, and the self-loops are not shown for clarity. Naturally, extinction (0) is an absorbing state.

Content requests: The number of requests for a content in a slot depends on its state in that slot, and given its popularity states, the numbers are independent across slots. In particular, r_1 and r_2 are the average number of requests in a slot for a content when it is in low (1) and high (2) popularity states, respectively.

Content caching: The BS has a *cache* where it can store up to K contents. For $s \in \{1,2\}$, let $n_{0,s}(t)$ and $n_{1,s}(t)$ denote the number of uncached and cached contents at the beginning of slot t that were in state s during $[t-1,t]$. Uncached contents can be *proactively* cached at slot boundaries while evicting other contents to meet the cache capacity constraints. We use $k_s(t)$ to denote the number of cached contents in state s after the execution of caching decisions at the beginning of slot t. Clearly, $k_s(t) \leq \sum_{c\in\{0,1\}} n_{c,s}(t)$ for $s \in \{1,2\}$ and $k_1(t) + k_2(t) \leq K$ for all $t \geq 1$. We let $\mathcal{N}(t)$ and $\mathcal{K}(t)$ denote the set of feasible content status and decisions, respectively, at time t. We illustrate various parameters in Figure 2.

1) Costs: We consider the following costs.

Caching cost: Contents' caching status are updated keeping their popularity and status in the previous slot in view. We let d denote the *cost of fetching* a content from its server and caching it. Clearly, the caching cost in slot t equals $d\sum_{s\in\{1,2\}}(k_s(t) - n_{1,s}(t))^+$.[1]

Content missing cost: If a content is not cached, unit cost per request is incurred. Notice that if a content is in state $s \in \{1,2\}$ during $[t-1,t]$ then it is requested expectedly $\bar{r}_s \coloneqq \sum_{s'\in\{1,2\}} p_{ss'}r_{s'}$ times in $[t,t+1]$. Consequently, the *expected content missing cost* in slot t equals $\sum_{s\in\{1,2\}}(\sum_{c\in\{0,1\}} n_{c,s}(t) - k_s(t))\bar{r}_s$.

We can express the total expected cost in a slot as a function of content status $n \coloneqq (n_{c,s}, c \in \{0,1\}, s \in \{1,2\})$ and caching decisions $k \coloneqq (k_1, k_2)$. Let $g(n,k)$ denote this cost. Then

$$g(n,k) = d\sum_{s\in\{1,2\}}(k_s - n_{1,s})^+ + \sum_{s\in\{1,2\}}\left(\sum_c n_{c,s} - k_s\right)\bar{r}_s. \tag{1}$$

[1] $r^+ \coloneqq \max\{r, 0\}$.

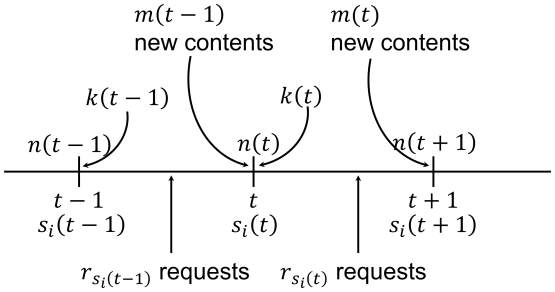


Fig. 2: Illustration of various parameters associated with the system. Content i, having popularity status $s_i(t-1)$ during $[t-1,t]$, draws on an average $r_{s_i(t-1)}$ requests in this period.The caching decisions $k(t) = (k_1(t), k_2(t))$ at the beginning of slot t depend on the state $n(t) = (n_{c,s}(t), c = 0,1, s = 1,2)$.

B. Optimal Caching Problem

Our objective is to make caching decisions so as to minimize the long-term time average cost. More precisely, given the initial content status n, we aim to solve the following problem.

$$\text{Minimize} \quad \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} g(n(t),k(t))\middle| n(1) = n\right] \tag{2}$$
$$\text{subject to} \quad k(t) \in \mathcal{K}(t) \ \forall\ t \geq 1.$$

1) Markov Decision Problem: We formulate the optimal caching problem as an average cost Markov decision problem. The slot boundaries are the decision epochs. The state of the system at decision epoch t is given by $n(t)$. We consider $k(t)$ to be the action at decision epoch t.

We define an operator $L : \mathbb{Z}_+^4 \times \mathbb{Z}_+^2 \to \mathbb{Z}_+^4$ for parsimonious presentation. For any tuple $(n,k) \in \mathbb{Z}_+^4 \times \mathbb{Z}_+^2$, for all $(c,s) \in \{0,1\} \times \{1,2\}$,

$$L_{c,s}(n,k) = \begin{cases} \sum_{s'} B\left(k_{s'}, p_{s's}\right) & \text{if } c = 1 \\ \sum_{s'} B\left(\sum_{c'} n_{c',s'} - k_{s'}, p_{s's}\right) & \text{otherwise,} \end{cases}$$

where $B(l,q)$ is the binomial random variable with parameters l and q. Given content status n and caching decision k at the beginning of a slot, Ln represents content status at the end of the same slot. From the description of the system model in Section II-A, given $n(t)$ and $k(t)$, the state at decision epoch $t+1$ is $n(t+1)$ with

$$n_{c,s}(t+1) = \begin{cases} L_{c,s}(n(t),k(t)) & \text{if } c = 1 \\ L_{c,s}(n(t),k(t)) + m_s(t) & \text{otherwise.} \end{cases} \tag{3}$$

For a state action pair (n,k), the expected single state cost is given by $g(n,k)$ as defined in Section II-A. A policy π is a sequence of mappings $\{u_t^\pi, t = 1,2,\cdots\}$ where $u_t^\pi : \mathcal{N}(t) \to \mathcal{K}(t)$. The cost of a policy π for an initial state n is

$$V^\pi(n) \coloneqq \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} g(n(t),u_t^\pi(n(t)))\middle| n(1) = n\right]$$

Let Π be the set of all policies. Then the optimal caching problem is $\min_{\pi\in\Pi} V^\pi(n)$.

ISBN 978-3-903176-55-3 © 2023 IFIP

Definition 1. (*Stationary Policies*) A policy $\pi = \{u_t^\pi, t = 1, 2, \cdots\}$ is called stationary if u_t^π are identical, say u, for all t. For brevity, we refer to such a policy as the stationary policy u. Following this, the content caching problem assumes an optimal stationary policy.

Restless multi-armed bandit formulation: The above Markov decision problem suffers from *curse of dimensionality*. But we make the following observations.

1) Contents' status evolves independently. Hence, the costs associated with the contents are linked only via the constraint on the caching actions, i.e., the capacity constraint of the cache.
2) The total cost can be decomposed into costs associated with different contents.

We thus see that the optimal caching problem is an instance of the *restless multi-armed bandit* (RMAB) problem with each content representing an arm. We show in Section III that this problem is *indexable*. This allows us to develop a Whittle index policy for the joint caching problem.

III. Whittle Index Policy

We outline our approach here. We first solve certain caching problems associated with each of the contents. We argue that these problems are indexable and provide explicit Whittle indices. The Whittle index for a state of content measures how rewarding it is to cache that content at that particular state. The Whittle index policy chooses those K contents whose current states have the largest Whittle indices and among these caches the ones with positive Whittle indices.

A. Single Content Caching Problem

We consider a Markov decision problem associated with Content i. Recall that the costs, and consequently, the caching decisions for the contents in a slot depend on their existing caching and popularity status (see Section II-B). Hence, in the MDP associated with the Content i, state and action in slot t are taken to be $x_i(t) \coloneqq (a_i(t-1), s_i(t-1))$ and $a_i(t)$, respectively, where $s_i(t)$ and $a_i(t)$ are the popularity status and caching decision, respectively, for content i in slot t. Its state and action spaces are $\{0,1\}\times\{0,1,2\}$ and $\{0,1\}$, respectively. Given $x_i(t) = (\bar{a}_i, \bar{s}_i)$ and $a_i(t) = a_i$, the state evolves as shown in Figure 1. The expected single-stage cost is

$$g_{i,w}(\bar{a}_i, \bar{s}_i, a_i) = wa_i + d(a_i - \bar{a}_i)^+ + (1 - a_i)\bar{r}_{\bar{s}_i}. \quad (4)$$

Observe that a constant penalty w is incurred in each slot in which Content i is stored in the cache. Unlike Section II-B, the single content MDP is a stochastic shortest path problem, states $(a_i, 0)$ for $a_i \in \{0,1\}$ being the terminal states. Here, a policy π is a sequence $\{u_t^\pi, t = 1, 2, \cdots\}$ where $u_t^\pi : \{0,1\} \times \{0,1,2\} \to \{0,1\}$. Given initial state (a_i, s_i) the problem minimizes

$$V_{i,w}^\pi(a_i, s_i) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} g_i(a_i(t), s_i(t), u_t^\pi(a_i(t), s_i(t)) \,\middle|\, a_i(0) = a_i, s_i(0) = s_i\right] \quad (5)$$

TABLE I: Optimal Actions ($a \in \{0,1\}, s' \neq s$)

Cases	$(u^*(a,s), u^*(a,s'))$
$w > \max\{\bar{r}_1, \bar{r}_2\}$	$(0,0)$
$\max\left\{\bar{r}_{s'} - (1-p_{s's'})d, \frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{1-p_{s's'}+p_{ss'}}\right\} < w < \bar{r}_{s'}$	$(0,a)$
$\bar{r}_s + p_{ss'}d < w < \bar{r}_{s'} - (1-p_{s's'})d$	$(0,1)$
$\max\{\bar{r}_1, \bar{r}_2\} < w < \min\left\{\frac{(1-p_{11})\bar{r}_2 + p_{21}\bar{r}_1}{1-p_{11}+p_{21}}, \frac{(1-p_{22})\bar{r}_1 + p_{12}\bar{r}_2}{1-p_{22}+p_{12}}\right\}$	(a,a)
$\bar{r}_s - p_{s0}d < w < \min\{\bar{r}_s + p_{ss'}d, \tilde{r}_s\}$	$(a,1)$
$w < \min\{\bar{r}_1 - p_{10}d, \bar{r}_2 - p_{20}d\}$	$(1,1)$

over all the policies to yield the optimal cost function $V_{i,w}(a_i, s_i) \coloneqq \min_\pi V_{i,w}^\pi(a_i, s_i)$. We analyze this problem below. However, we omit the index i for brevity.

Bellman's Equations: The value function $V : \{0,1\} \times \{0,1,2\} \to \mathbb{R}_+$ satisfies the following Bellman's equations.

$$V(a,0) = 0,$$
$$V(a,s) = \min_{a'}\Big\{wa' + d(a'-a)^+ + (1-a')\bar{r}_s + \sum_{s'} p_{ss'}V(a',s')\Big\},$$

for all $a \in \{0,1\}$ and $s \in \{1,2\}$.

1) The Optimal Policy: The single content caching problem also assumes an optimal stationary policy. Recall that $\bar{r}_s \coloneqq \sum_{s' \in \{1,2\}} p_{ss'} r_{s'}$ for $s \in \{1,2\}$. We further define

$$\tilde{r}_s \coloneqq \frac{(1-p_{ss})\bar{r}_{s'} + p_{s's}\bar{r}_s}{1 - p_{ss} + p_{s's}} - \frac{(1-p_{s's})(1-p_{ss}) - p_{ss'}p_{s's}}{1 - p_{ss} + p_{s's}}d \quad (6)$$

for $(s, s') = (1,2)$ or $(2,1)$. Then the optimal policy is given by the following theorem.

Theorem 1. *For $a \in \{0,1\}$ and $(s,s') \in \{(1,2),(2,1)\}$, the optimal actions $(u^*(a,s), u^*(a,s'))$ are as given in Table I.*

Proof: See Appendix A in [36]. ∎

In Table I we have divided the values of penalty parameter w in six different intervals for the purpose of determining the optimal policy. The expressions for these intervals can be simplified, and a few of these may even be empty, depending on the parameter values. We omit this analysis here due to space constraints. Please see Append A [36] for details.

B. Indexability of the RMAB

From the structure of the optimal policy in Table I we can deduce that it is of threshold type. In particular, there exist $(w(a,s), a = 0,1, s = 1,2)$ such that

$$u^*(a,s) = \begin{cases} 1 & \text{if } w < w(a,s) \\ 0 & \text{otherwise.} \end{cases}$$

The values of $(w(a,s), a = 0,1, s = 1,2)$ depend on various parameters. Clearly, the RMAB is indexable, and $w(a,s)$s are the Whittle indices for various states. We present $(w(a,s), w(a,s'))$ for $a \in \{0,1\}$ and $(s,s') \in \{(1,2),(2,1)\}$ in Table II where have assumed, without any loss of generality,

ISBN 978-3-903176-55-3 © 2023 IFIP

that $\bar{r}_{s'} \geq \bar{r}_s$. We formalize these assertions in the following theorem.

Theorem 2. *The content caching problem is indexable. For $a \in \{0,1\}$ and $(s,s') \in \{(1,2),(2,1)\}$, if $\bar{r}_{s'} \geq \bar{r}_s$, the Whittle indices are as in Table II.*

Proof: For details, see Appendix B, [36]. ∎

Consider the special but reasonable scenario where the probability distribution (p_{22}, p_{21}, p_{20}) is stochastically larger than (p_{12}, p_{11}, p_{10}). In this scenario $\bar{r}_2 \geq \bar{r}_1$ and $p_{20} \leq p_{10}$ implying that $\bar{r}_2 - \bar{r}_1 \geq (p_{20} - p_{10})d$. Hence the Whittle indices are given by the first two cases of Table II. It can can easily verified that $w(0,1) \leq w(0,2) \leq w(1,2)$ and $w(0,1) \leq w(1,1) \leq w(1,2)$ as expected in this case.

C. Whittle Index Policy for the RMAB

We now describe the Whittle index policy for the joint content caching problem. As stated earlier, it chooses those K arms whose current states have the largest Whittle indices and among these caches the ones with positive Whittle indices. In the content caching problem at hand, each content for which we have to make a caching decision can be in one of the four states $(a,s), a = 0,1, s = 1,2$. We sort $w(a,s)$s and, subject to the cache capacity, cache the contents with the highest Whittle index if it is positive, then cache the contents with the second highest Whittle index if it is positive and so on.

Holding cost: There can also be a holding cost for keeping content in the cache. Let h denote this fixed holding cost per content per slot. The content caching problem remains unchanged except that single stage cost associated with Content i becomes

$$g_{i,w}(\bar{a}_i, \bar{s}_i, a_i) = ha_i + d(a_i - \bar{a}_i)^+ + (1 - a_i)r_{\bar{s}_i}.$$

Comparing it with (4), we see that the penalty w can be interpreted as the fixed holding cost. Obviously, in the presence of the holding cost, the content caching problem can be solved following the same approach as above.

IV. Performance of the Whittle Index Policy

We analyze the performance of the proposed Whittle index policy in this section. Following the approach in [4], we consider a relaxed version of the RMAB problem and derive the associated optimal average cost, which is a lower bound on the optimal cost of the RMAB problem in Section II-B, i.e., a lower bound on $\min_{\pi \in \Pi} V^{\pi}(n)$. We then show that the average cost of the Whittle index policy in Section III-C is close to this lower bound and hence to the optimal cost of the RMAB problem in Section II-B.

We now constrain the average number of cached contents to be less than or equal to K. Caching K or fewer contents each time is one way to satisfy this constraint. We can formalize this relaxed content caching problem as follows. Let $\mathcal{C}_s(t), s \in \{1,2\}$ denote the set of contents arrived until t and having initial state s, and $\mathcal{C}(t) = \mathcal{C}_1(t) \cup \mathcal{C}_2(t)$. Further, for any content $i \in \mathcal{C}(t)$, let A_i and D_i denote its arrival and exit times, respectively. Then the time-average cost can be expressed as

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{i \in \mathcal{C}(T)} \sum_{t=A_i}^{(D_i-1)\wedge T} c_i(s_i(t), a_i(t)) \middle| n(1) = n \right].$$

We aim at minimizing the above cost subject to

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{i \in \mathcal{C}(T)} \sum_{t=A_i}^{(D_i-1)\wedge T} a_i(t) \middle| n(1) = n \right] \leq K.$$

The following theorem gives the optimal policy for this relaxed problem in terms of the Whittle indices.

Theorem 3. *Let $\bar{V}$ denote the optimal average cost of the above RMAB problem. Then*

$$\bar{V} = \max_{w \geq 0} \{\lambda_1 V_w(0,1) + \lambda_2 V_w(0,2) - wK\} \tag{7}$$

where where $V_w(0,1)$ and $V_w(0,2)$ are the value functions of the arms as defined in Section III-A. This optimal value is achieved by a policy that caches at each time the contents whose current Whittle indices exceed a threshold w^*, w^* being the value that achieves the maximum in (7)*,*

Proof: The proof is based on a Lagrangian duality. For details, see Appendix C [36].

∎

We have provided the optimal policies as functions of the penalty parameter w in Table I. We can use these optimal policies to compute $V_w(0,1)$ and $V_w(0,2)$ for various w. We present these value functions in Table III.

Remark 1. Let V and V^W denote, respectively, the average cost of the optimal policy and that of the Whittle index policy for the original RMAB problem with the strict activation constraint. It is easy to see that

$$\bar{V} \leq V \leq V^W$$

i.e., $\bar{V}$ as specified in (7) provides a lower bound on the optimal cost. It can be used as a benchmark for gauging the performance of the Whittle index policy for the original RMAB problem.

Asymptotic optimality: The Whittle index policy performs close to the optimal policy, especially when the average number of contents in the system is high. However, a formal assessment of the performance of the Whittle index policy has evaded us until now. We numerically demonstrate the performance of various policies in the next section.

Greedy policy: We have assumed that contents' popularity and extinction evolve independently of the caching decisions. In the absence of caching costs (switching costs), a greedy policy that only accounts for single-stage costs would be optimal for the content caching problem. On the other hand, in the presence of caching costs, the state evolution of the MDP depends on the caching decisions. In this case, the greedy policy performs abysmally.

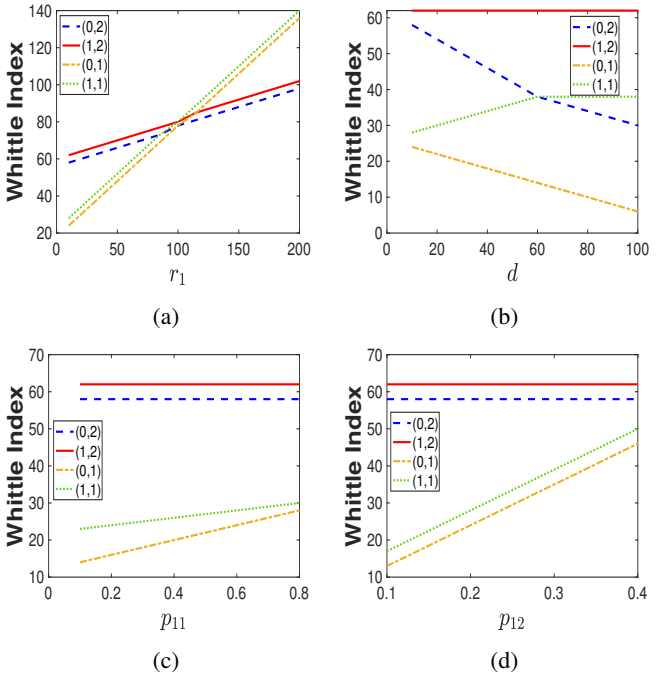
ISBN 978-3-903176-55-3 © 2023 IFIP

TABLE II: Whittle Indices ($s' \neq s, \bar{r}_{s'} \geq \bar{r}_s$)

Cases	$w(0,s)$	$w(1,s)$	$w(0,s')$	$w(1,s')$
$\bar{r}_{s'} - \bar{r}_s \geq \max\{p_{s'0} - p_{s0}, 1 - p_{s's'} + p_{ss'}\}d$	$\bar{r}_s - p_{s0}d$	$\bar{r}_s + p_{ss'}d$	$\bar{r}_{s'} - (1 - p_{s's'})d$	$\bar{r}_{s'}$
$(p_{s'0} - p_{s0})d \leq \bar{r}_{s'} - \bar{r}_s < (1 - p_{s's'} + p_{ss'})d$	$\bar{r}_s - p_{s0}d$	$\frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{1-p_{s's'}+p_{ss'}}$	$\tilde{r}_s$	$\bar{r}_{s'}$
$\bar{r}_{s'} - \bar{r}_s < (p_{s'0} - p_{s0})d$	$\tilde{r}_{s'}$	$\frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{1-p_{s's'}+p_{ss'}}$	$\bar{r}_{s'} - p_{s'0}d$	$\bar{r}_{s'}$

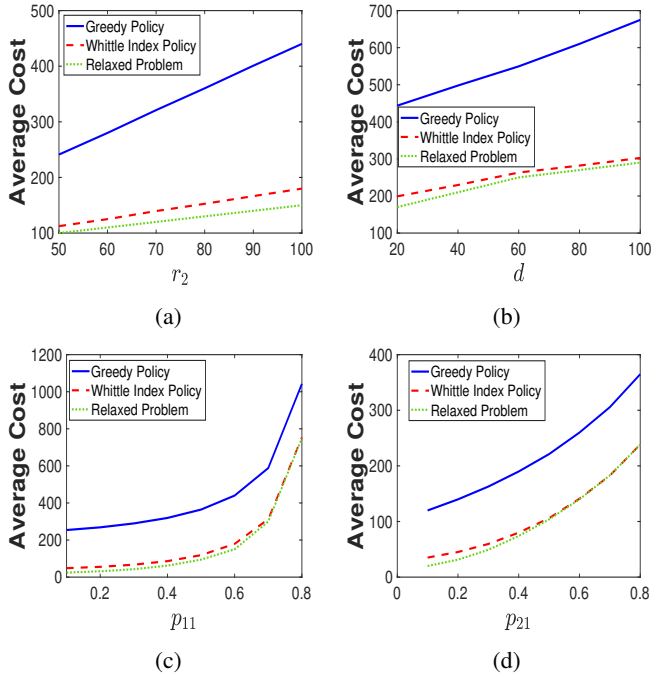
TABLE III: $V_w(0,s)$ and $V_w(0,s')$

Cases	$V_w(0,s)$	$V_w(0,s')$
$w \geq \max\{\bar{r}_1, \bar{r}_2\}$	$\frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$	$\frac{(1-p_{ss})\bar{r}_{s'} + p_{s's}\bar{r}_s}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$
$\max\left\{\bar{r}_{s'} - (1-p_{s's'})d, \frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{1-p_{s's'}+p_{ss'}}\right\} \leq w < \bar{r}_{s'}$	$\frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$	$\frac{(1-p_{ss})\bar{r}_{s'} + p_{s's}\bar{r}_s}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$
$\bar{r}_s + p_{ss'}d \leq w < \bar{r}_{s'} - (1-p_{s's'})d$	$\frac{(1-p_{s's'})(\bar{r}_s + p_{ss'}d) + p_{ss'}w}{(1-p_{ss})(1-p_{s's'}) - p_{ss'}p_{s's}}$	$\frac{p_{s's}\bar{r}_s + (1-p_{s's'})((1-p_{s's'})d + w)}{(1-p_{ss})(1-p_{s's'}) - p_{ss'}p_{s's}}$
$\max\{\tilde{r}_1, \tilde{r}_2\} \leq w < \min\left\{\frac{(1-p_{11})\bar{r}_2 + p_{21}\bar{r}_1}{1-p_{11}+p_{21}}, \frac{(1-p_{22})\bar{r}_1 + p_{12}\bar{r}_2}{1-p_{22}+p_{12}}\right\}$	$\frac{(1-p_{s's'})\bar{r}_s + p_{ss'}\bar{r}_{s'}}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$	$\frac{(1-p_{ss})\bar{r}_{s'} + p_{s's}\bar{r}_s}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$
$\bar{r}_s - p_{s0}d \leq w < \min\{\bar{r}_s + p_{ss'}d, \tilde{r}_s\}$	$\frac{\bar{r}_s}{(1-p_{ss})} + \frac{p_{ss'}(w+d)}{(1-p_{ss})} + \frac{p_{ss'}}{(1-p_{ss})}\frac{((1-p_{ss})p_{s's} + p_{s's}p_{ss'})w}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$	$w + d + \frac{((1-p_{ss})p_{s's'} + p_{s's} + p_{s's}p_{ss'})w}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$
$w < \min\{\bar{r}_1 - p_{10}d, \bar{r}_2 - p_{20}d\}$	$w + d + \frac{((1-p_{s's'})p_{ss} + p_{ss'}(1+p_{s's}))w}{(1-p_{s's'})(1-p_{ss}) - p_{ss'}p_{s's}}$	$w + d + \frac{((1-p_{ss})p_{s's'} + p_{s's}(1+p_{ss'}))w}{(1-p_{s's'})(1-p_{ss'}) - p_{ss'}p_{s's}}$



(a) (b) (c) (d)

Fig. 3: Whittle indices vs r_1, d, p_{11} and p_{12}. We use $p_{22} = 0.6$, $p_{21} = 0.2$ and $r_2 = 100$. Figure 3a gives the Whittle indices vs r_1 for $d = 10$, $p_{11} = 0.6$ and $p_{12} = 0.2$. Figure 3b gives the Whittle indices vs d when the the parameters are $r_1 = 10$, $p_{11} = 0.6$ and $p_{12} = 0.2$. Figure 3c gives the Whittle indices vs p_{11} for $r_1 = 10$, $d = 10$, and $p_{12} = 0.2$. Figure 3d gives the Whittle indices vs p_{11} when the parameters are $r_1 = 10$, $d = 10$ and $p_{11} = 0.6$.



(a) (b) (c) (d)

Fig. 4: Average cost vs r_2, d, p_{22} and p_{12}. We use $p_{11} = 0.6$, $p_{21} = 0.2$, $r_1 = 10$, $\lambda_1 = 1$, $\lambda_1 = 2$ and $K = 5$. Figure 4a gives Average Cost vs r_2 for $d = 10$, $p_{22} = 0.6$ and $p_{12} = 0.2$. Figure 4b gives Average Cost vs d for $r_2 = 100$ and $p_{22} = 0.6$ and $p_{12} = 0.2$. Figure 4c gives Average cost vs p_{22} for $r_2 = 100$ and $d = 10$ and $p_{12} = 0.2$. Figure 4d gives Average cost vs p_{12} for $r_2 = 100$ and $d = 10$, $p_{11} = 0.2$ and $p_{22} = 0.6$.

V. Numerical Results

A. Whittle indices vs system parameters

Whittle indices vs r_1: We plot Whittle indices vs r_1, for a single content at states $(1,2)$, $(0,2)$, $(1,1)$ and $(0,1)$. In Figure 3a, we observe that the Whittle index for any state is a piecewise linear increasing function in r_1. This is true

ISBN 978-3-903176-55-3 © 2023 IFIP

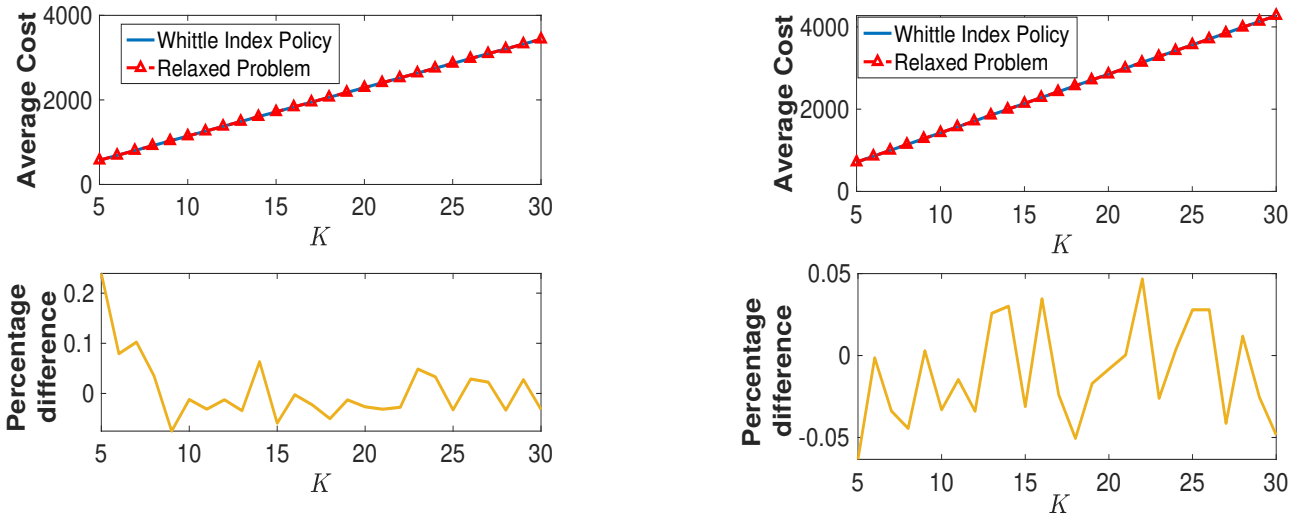


Fig. 5: Average cost vs K. We use $p_{11} = 0.6$, $p_{22} = 0.6$, $p_{12} = 0.2$, $p_{21} = 0.2$, $r_1 = 10$ and $r_2 = 100$. $\lambda_1 = 1.25$ and $\lambda_2 = 2.5$ at $K = 5$. In Figure 5a and Figure 5b, we use $d = 10$ and $d = 100$, respectively.

because we prefer to cache contents that see a lot of requests. Therefore, as r_1 increases, the Whittle index also increases so that contents are cached even if we incur more holding costs.

Whittle indices vs d: We plot Whittle indices vs d, for a single content at states $(1,2)$, $(0,2)$, $(1,1)$ and $(0,1)$. We observe that the Whittle index at the state $(1,2)$ is constant. This is true because, based on the parameters that we have chosen, the Whittle index at $(1,2)$ is independent of d. Based on the parameters that we have chosen, the Whittle index at $(0,2)$ and $(1,1)$ is a piece-wise linear decreasing and piece-wise linear increasing function of d, respectively. Whittle index at $(0,1)$ goes down linearly with d.

Whittle indices vs transition probabilities: Figure 3 gives Whittle indices vs transition probabilities. Based on the parameters that we have chosen, the Whittle index at $(0,1)$ and $(1,1)$ increase linearly with transition probability. Since the Whittle index at $(1,2)$ and $(0,2)$ are independent of the transition probability, we observe that the Whittle index at the state $(1,2)$ and $(0,2)$ is constant.

B. Average Cost vs system parameters

We compare the average relaxed cost, average cost under Whittle index Policy and greedy policy against various parameters as we see in Figure 4. In Greedy policy, at each time slot, we decide to cache up to K contents, such that a content i is cached if it happens to be in the state $x_g \in \{0,1\} \times \{1,2\}$ such that

$$x_g = \underset{x_i \in \{0,1\} \times \{1,2\}}{\operatorname{argmin}} \{g_i(x_i,1) - g_i(x_i,0) < 0\}.$$

In each time slot, if more than K contents happen to be in state x_g, then the ties are broken arbitrarily. From Figures 4a, 4b, 4c and 4d, we observe that the Whittle index policy performs better than the greedy policy. We also observe that the Relaxed RMAB has a slight edge over the Whittle index policy.

Average cost increases with the content request rate. This is true because, in the event of a cache miss, a higher request rate implies a higher service cost. This leads to an increase in average cost. In the greedy policy, we face more cache misses than in the Whittle index policy. Therefore, as we see in Figure 4a, as r_2 increases, the Greedy policy performs worse than the Whittle index policy. As the fetching cost d increases, the cost of caching content increases. In Figure 4b, as d increases, the Whittle index policy performs better than the Greedy policy.

From Figures 4c and 4d we observe that the average cost increases with the transition probability p_{22} and p_{12}, respectively. This is true because as the transition probability increases, contents tend to stay in the system forever. As p_{22} increases, the missing cost associated with the high-popularity content increases, leading to a higher average cost. Similarly, as p_{12} increases, the missing cost associated with the low-popularity content increases, leading to a higher average cost.

In Figures 4c and 4d, Whittle index policy performs as good as the Relaxed RMAB, as the transition probability increases. This is true because as the transition probability increases, contents tend to stay in the system forever. Whittle index policy performs close to the optimal policy as the mean number of contents at the server increases.

We also examined the performance of the Whittle index policy as we varied the number of cached contents, denoted as K. In practical scenarios, the caching problem is meaningful when the number of cached contents is less than the average number of contents at the server. To this end, the arrival rate of content of popularity state s at K is computed as follows.

$$\lambda_s(K) = \lambda_s(K-1)\frac{K}{K-1}, s \in \{1,2\}.$$

At $K = 5$, we used $\lambda_1 = 1.25$ and $\lambda_2 = 2.5$. Figures 5a and

ISBN 978-3-903176-55-3 © 2023 IFIP

5b illustrate that the Whittle index policy performs comparably to the Relaxed RMAB policy.

VI. Conclusion and Future Work

We considered optimal caching of dynamic contents. We cast the problem as an average cost Markov decision problem and showed that it is an instance of RMABs. We established that the caching problem is indexable and obtained explicit expressions for the Whittle indices. Furthermore, we showed that the Whittle Index Policy outperforms the greedy policy and performs comparably to the relaxed RMAB policy.

We plan to extend our findings to more than two popularity levels in our future work. Additionally, we aim to apply our approach to a distributed caching system that involves multiple base stations. We also intend to integrate a Reinforcement Learning-based algorithm that can handle scenarios where the popularity dynamics are unknown, making it more applicable to real-world settings.

Acknowledgements

Anu Krishna and Chandramani Singh acknowledge the supports from the MeitY's Visvesvaraya PhD Scheme and Aircel TCoE project 39010C, respectively.

References

[1] W. Chai and K. Casley, "Software as a service (saas)," https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service, 2022.

[2] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: A view from the edge," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 15–28.

[3] B. Tan and L. Massoulié, "Optimal content placement for peer-to-peer video-on-demand systems," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 566–579, 2013.

[4] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of Applied Probability*, vol. 25, pp. 287–298, 1988.

[5] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queueing network control," in *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*. IEEE, 1994, pp. 318–322.

[6] P. Mansourifard, T. Javidi, and B. Krishnamachari, "Optimality of myopic policy for a class of monotone affine restless multi-armed bandits," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 877–882.

[7] V. Raghunathan, V. Borkar, M. Cao, and P. R. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 2008, pp. 1570–1578.

[8] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Minimizing the age of information in broadcast wireless networks," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 844–851.

[9] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *Journal of Applied Probability*, vol. 27, no. 3, pp. 637–648, 1990.

[10] Q. Zhao, "Multi-armed bandits: Theory and applications to online learning in networks," in *Multi-Armed Bandits*, 2019.

[11] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[12] R. S. Prakash, N. Karamchandani, V. Kavitha, and S. Moharir, "Partial service caching at the edge," in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2020, pp. 1–8.

[13] N. Carlsson and D. Eager, "Ephemeral content popularity at the edge and implications for on-demand caching," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 6, pp. 1621–1634, 2017.

[14] J. Gao, S. Zhang, L. Zhao, and X. Shen, "The design of dynamic probabilistic caching with time-varying content popularity," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1672–1684, 2021.

[15] L. Maggi, L. Gkatzikis, G. Paschos, and J. Leguay, "Adapting caching to audience retention rate: Which video chunk to store?" *arXiv preprint arXiv:1512.03274*, 2015.

[16] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1111–1125, 2018.

[17] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal dynamic proactive caching via reinforcement learning," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[18] Z. Zhang, X. Wei, C.-H. Lung, and Y. Zhao, "icache: An intelligent caching scheme for dynamic network environments in icn-based iot networks," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1787–1799, 2023.

[19] B. N. Bharath, K. G. Nagananda, D. Gündüz, and H. V. Poor, "Caching with time-varying popularity profiles: A learning-theoretic perspective," *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 3837–3847, 2018.

[20] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Single vs distributed edge caching for dynamic content," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 669–682, 2022.

[21] P. Kaswan, M. Bastopcu, and S. Ulukus, "Timely cache updating in parallel multi-relay networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[22] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh, "Fresh caching of dynamic content over the wireless edge," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2315–2327, 2022.

[23] V. S. C. L. Narayana, M. Agarwala, N. Karamchandani, and S. Moharir, "Online partial service hosting at the edge," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–9.

[24] S. Krishnendu, B. N. Bharath, N. Garg, V. Bhatia, and T. Ratnarajah, "Learning to cache: Federated caching in a cellular network with correlated demands," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1653–1665, 2022.

[25] N. Carlsson and D. Eager, "Optimized dynamic cache instantiation and accurate lru approximations under time-varying request volume," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 779–797, 2023.

[26] N. Mhaisen, G. Iosifidis, and D. Leith, "Online caching with optimistic learning," in *2022 IFIP Networking Conference (IFIP Networking)*, 2022, pp. 1–9.

[27] S. Fan, I. Hou, V. S. Mai, and L. Benmohamed, "Online service caching and routing at the edge with switching cost," *CoRR*, vol. abs/2107.10446, 2021. [Online]. Available: https://arxiv.org/abs/2107.10446

[28] V. C. L. Narayana, S. Jain, and S. Moharir, "Caching partial files for content delivery," in *2019 National Conference on Communications (NCC)*, 2019, pp. 1–6.

[29] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 177–186.

[30] A. Krishna, R. Burra, and C. Singh, "Caching dynamic contents with varying popularity," in *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 2021, pp. 1–8.

[31] P. Whittle, "Arm-Acquiring Bandits," *The Annals of Probability*, vol. 9, no. 2, pp. 284 – 292, 1981.

[32] G. Weiss, "Branching bandit processes," *Probability in the Engineering and Informational Sciences*, vol. 2, no. 3, p. 269–278, 1988.

[33] D. Chakrabarti, R. Kumar, F. Radlinski, and E. Upfal, "Mortal multi-armed bandits," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2008.

[34] G. Xiong, S. Wang, G. Yan, and J. Li, "Reinforcement learning for dynamic dimensioning of cloud caches: A restless bandit approach," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 2108–2117.

[35] A. Madnaik, S. Moharir, and N. Karamchandani, "Renting edge computing resources for service hosting," 2022.

[36] A. Krishna and C. Singh, "Caching dynamic contents via mortal restless bandits," 2023. [Online]. Available: https://tinyurl.com/2s3brrye