Caching Contents with Varying Popularity Using Restless Bandits

K. J. Pavamana(✉) and Chandramani Singh

Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru, India
{pavamanak,chandra}@iisc.ac.in

Abstract. We study content caching in a wireless network in which the users are connected through a base station that is equipped with a finite capacity cache. We assume a fixed set of contents whose popularity vary with time. Users' requests for the contents depend on their instantaneous popularity levels. Proactively caching contents at the base station incurs a cost but not having requested contents at the base station also incurs a cost. We propose to proactively cache contents at the base station so as to minimize content missing and caching costs. We formulate the problem as a discounted cost Markov decision problem that is a restless multi-armed bandit problem. We provide conditions under which the problem is indexable and also propose a novel approach to manoeuvre a few parameters to render the problem indexable. We demonstrate efficacy of the Whittle index policy via numerical evaluation.

Keywords: Caching · Restless bandits · Threshold policy · Whittle index

1 Introduction

The exponential growth of intelligent devices and mobile applications poses a significant challenge to Internet backhaul, as it struggles to cope with the surge in traffic. According to Cisco's annual Internet report 2023 [1], approximately two-thirds of the world's population will have access to Internet by 2023, and the number of devices connected to IP networks will exceed three times the global population. This extensive user base will generate a high demand for multimedia content, such as videos and music. However, this increased traffic is often due to repeated transmissions of popular content, leading to an unnecessary burden on the network. The resulting influx of content requests has adverse effects on latency, power consumption, and service quality.

To address these challenges, proactive content caching at the periphery of mobile networks has emerged as a promising solution. By implementing caches

This work was supported by Centre for Network Intelligence, Indian Institute of Science (IISc), a CISCO CSR initiative.

E. Kalyvianaki and M. Paolieri (Eds.): VALUETOOLS 2023, LNICST 539, pp. 133–150, 2024.
https://doi.org/10.1007/978-3-031-48885-6_9

at base stations, it becomes possible to pre-store requested content in advance. As a result, content requests can be efficiently served from these local caches instead of remote servers, benefiting both users and network operators. Users experience reduced latency and improved quality of experience when accessing content from intermediary base stations. For network operators, caching content at the network edge significantly reduces network overhead, particularly in cases where multiple users request the same content, such as popular videos and live sports streams.

Notwithstanding the widespread benefits of including content caching abilities in the networks, there are also several challenges in deploying the caching nodes. First of all, the size of a cache is constrained and caching contents incurs a cost as well. So, it is not viable to store each and every content that can possibly be requested by the user in the cache. This calls for efficient strategies to determine the contents that should be stored in the cache.

In this work, we aim at minimizing the discounted total cost incurred in delivering contents to end users, which consists of content missing and caching costs. We consider a fixed set of contents with varying popularity and a single cache, and design policies that decide which contents should be cached so as to minimize the discounted total cost while simultaneously satisfying caching capacity constraint of the base station.

The problem at hand is framed as a Markov decision process (MDP) [2], resembling a restless multiarmed bandit (RMAB) scenario [3]. Although value iteration [2,4] theoretically solves RMAB, it is plagued by the curse of dimensionality and provides limited solution insights. Therefore, it is advantageous to explore less intricate approaches and assess their efficacy. An esteemed strategy for RMAB problems is the Whittle index policy [3]. This Whittle index policy has been widely employed in the literature and has proven highly effective in practical applications [5–7]. Whittle [3] demonstrated the optimality of index-based policies for the Lagrangian relaxation of the restless bandit problem, introducing the concept of the Whittle index as a useful heuristic for restless bandit problems. Hence, we suggest employing this policy to address the task of optimizing caching efficiently.

1.1 Related Work

Content Caching There are two types of caching policies, proactive or reactive. Under a reactive policy, a content can be cached upon the user's request. When a user requests a specific content, the system first checks if the content is available in the local cache. If the content is found in the cache, it is delivered to the user directly from the cache. However, if the content is not present in the cache, the system initiates a process to fetch the content progressively from the server. Li et al. [8] proposed a reactive caching algorithm PopCaching that uses popularity evolution of the contents in determining which contents to cache.

In proactive caching, popularity prediction algorithms are used to predict user demand and to decide which contents are cached and which are evicted. Sadeghi et al. [9] proposed an intelligent proactive caching scheme to cache

fixed collection of contents in an online social network to reduce the energy expenditure in downloading the contents. Gao et al. [10] proposed a dynamic probabilistic caching for a scenario where contents popularity vary with time. Abani et al. [11] designed a proactive caching policy that relies on predictability of the mobility patterns of mobiles to predict a mobile device's next location and to decide which caching nodes should cache which contents. Traverso et al. [12] introduced a novel traffic model known as the Shot Noise Model (SNM). This parsimonious model effectively captures the dynamics of content popularity while also accounting for the observed temporal locality present in actual traffic patterns. ElAzzouni et al. [13] studied the impact of predictive caching on content delivery latency in wireless networks. They establish a predictive multicast and caching concept in which base stations (BSs) in wireless cells proactively multicast popular content for caching and local access by end users.

Restless Multi-armed Bandit Problems In a restless multi-armed bandit (RMAB) problem, a decision maker must select a subset of M arms from K total arms to activate at any given time. The controller has knowledge of the states and costs associated with each arm and aims to minimize the discounted or time-average cost. The state of an arm evolves stochastically based on transition probabilities that depend on whether the bandit is active. Solving an RMAB problem through dynamic programming is computationally challenging, even for moderately sized problems. Whittle [3] proposed a heuristic solution known as the Whittle index policy, which addresses a relaxed version of the RMAB problem where M arms are only activated on average. This policy calculates the Whittle indices for each arm and activates the M arms with the highest indices at each decision epoch. However, determining the Whittle indices for an arm requires satisfying a certain indexability condition, which can be generally difficult to verify.

Xiong et al. [14] have formulated a content caching problem as a RMAB problem with the objective being minimizing the average content serving latency. They established the indexability of the problem and used the Whittle index policy to minimize the average latency.

There are very few works on RMABs with switching costs, e.g., costs associated with switching active arms. Ny et al. [15] considered a RMAB problem with switching costs, but they allow only one bandit to be active at any time. Incorporating switching costs in RMAB problems makes the states of the bandits multidimensional. This renders calculation of the Whittle indices much more complex. The literature on multidimensional RMAB is scarce. The main difficulty lies in establishing indexability, i.e., in ordering the states in a multidimensional space. Notable instances are [16–18] in which the authors have derived Whittle indices. But none of them have considered switching cost. We pose the content caching problem as a RMAB problem with switching costs (it is called as caching cost in the context of caching problem) and develop the simple Whittle index policy.

Organisation The rest of the paper is organised as follows. In Sect. 2, we present the system model for the proactive content caching problem and formulate the

problem as a RMAB. In Sect. 3, we show that each single arm MDP has a threshold policy as the optimal policy and is indexable. In Sect. 4, we manoeuvre a few cost parameters to render the modified MDP indexable in a few special cases in which the original MDP is nonindexable. In Sect. 6, we show efficacy of the Whittle index policy via numerical evaluation. Finally, we outline future directions in Sect. 7.

2 System Model and Caching Problem

In this section, we first present the system model and then pose the optimal caching problem as a discounted cost Markov decision problem.

2.1 System Model

We consider a wireless network where the users are connected to a single base station (BS) which in turn is connected to content servers via the core network. The content providers have a set of K contents, $\mathcal{C} = \{1, 2, \ldots, K\}$ which are of equal size, at the servers. The BS has a *cache* where it can store up to M contents. We assume a slotted system. Caching decisions are taken at the slot boundaries. We use $a(t) = (a_i(t), i \in \mathcal{C})$ to denote the caching status of various contents at the beginning of slot t; $a_i(t) = 1$ if Content i is cached and $a_i(t) = 0$ otherwise. We let $\mathcal{A}$ denote the set of feasible status vectors;

$$\mathcal{A} = \left\{ a \in \{0,1\}^K : \sum_{i \in \mathcal{C}} a_i \leq M \right\}.$$

Content Popularity We assume that the contents' popularity is reflected in the numbers of requests in a slot and varies over time. For any content, its popularity evolution may depend on whether it is stored in the BS' cache or not. We assume that for any content, say for Content i, given its caching status a_i, the numbers of requests in successive slots evolve as a discrete time Markov chain as shown in Fig. 1.[1]

In Fig. 1, numbers of requests $\phi_r^i \in \mathbb{Z}_+$ for $r = 1, 2, \ldots$ and it is an increasing sequence. We do not show self loops for clarity. We also make the following assumption.

Assumption 1. For all $i \in \mathcal{C}$, $p_i^{(1)} \geq p_i^{(0)}$ and $q_i^{(1)} \leq q_i^{(0)}$.

Assumption 1 suggests that, statistically, a content's popularity grows more if it is cached. We need this assumption to establish that optimal caching policy is a *threshold policy*.

[1] There are several instances of content popularity being modelled as Markov chains, e.g., see [9,19,20].

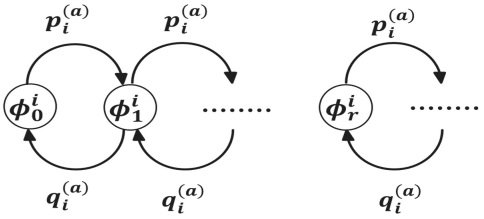


Fig. 1. Popularity evolution of a content. Given its caching status $a \in \{0, 1\}$, the average number of requests per slot vary in accordance with these transition probabilities. For clarity, self-loops are not shown.

Costs We consider the following costs.
Content missing cost If a content, say Content i, is not cached at the beginning of a slot and is requested ϕ_r times in that slot, a cost $C_i(\phi_r)$ is incurred. For brevity we write this cost as $C_i(r)$ with a slight abuse of notation. Naturally, the functions $C_i : \mathbb{Z}_+ \to \mathbb{R}_+$ are non-decreasing. We also make the following assumption.

Assumption 2. For all $i \in \mathcal{C}$,

1. $C_i : \mathbb{Z}_+ \to \mathbb{R}_+$ are non-decreasing,
2. $C_i : \mathbb{Z}_+ \to \mathbb{R}_+$ are concave, i.e.,

$$C_i(r+1) - C_i(r) \leq C_i(r) - C_i(r-1) \ \forall r \geq 1. \tag{1}$$

Caching cost At the start of each slot, we have the ability to adjust the caching status of the contents based on their requests in the previous slot. Specifically, we can *proactively* cache contents that are currently not cached, while removing other contents to ensure compliance with the cache capacity restrictions. Let's use d to represent the cost associated with fetching content from its server and caching it.

We can express the total expected cost in a slot as a function of the cache status in this and the previous slots, say a and $\bar{a}$, respectively, and the request vector in the previous slot, say $\bar{r}$. Let $c(\bar{a}, \bar{r}, a)$ denote this cost. Clearly, $c(\bar{a}, \bar{r}, a) = \sum_{i \in \mathcal{C}} c_i(\bar{a}_i, \bar{r}_i, a_i)$ where

$$\begin{aligned} c_i(\bar{a}_i, \bar{r}_i, a_i) =& d a_i(1 - \bar{a}_i) + (1 - a_i)\Big(p^{a_i} C_i(\bar{r}_i + 1) + \\ & q^{a_i} C_i((\bar{r}_i - 1)^+) + (1 - p^{a_i} - q^{a_i}) C_i(\bar{r}_i)\Big). \end{aligned} \tag{2}$$

We define operators $T^u, u \in \{0, 1\}$ for parsimonious presentation. For any function $g : \mathbb{Z}_+ \to \mathbb{R}_+$, for all $r \in \mathbb{Z}_+$,

$$T^u g(r) = p^u g(r+1) + q^u g((r-1)^+) + (1 - p^u - q^u) g(r).$$

We can rewrite (2) in terms of T^us;

$$c_i(\bar{a}_i, \bar{r}_i, a_i) = da_i(1 - \bar{a}_i) + (1 - a_i)\Big(T^{a_i} C_i(\bar{r}_i)\Big). \tag{3}$$

2.2 Optimal Caching Problem

Our goal is to determine caching decisions that minimize the long-term expected discount cost. To be more precise, when provided with the initial request vector r and cache state a, we aim to solve the following problem:

$$\begin{aligned} &\text{Minimize} \quad \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^t c(a(t-1), r(t-1), a(t)) \middle| \begin{matrix} a(0)=a, \\ r(0)=r \end{matrix}\right] \\ &\text{subject to} \quad a(t) \in \mathcal{A} \ \forall\ t \geq 1. \end{aligned} \tag{4}$$

The performance measure (4) is meaningful when the future costs are less important.[2]

Markov Decision Problem We formulate the optimal caching problem as a discounted cost Markov decision problem. The slot boundaries are the decision epochs. The state of the system at decision epoch t is given by the tuple $x(t) := (a(t-1), r(t-1))$. We consider $a(t)$ to be the action at decision epoch t. Clearly, the state space is $\mathcal{A} \times \mathbb{Z}_+^K$, and the action space is $\mathcal{A}$. From the description of the system in Sect. 2.1, given state $x(t) = (\bar{a}, \bar{r})$ and action $a(t) = a$, the state at decision epoch $t+1$ is $x(t+1) = (a, r)$ where

$$r_i = \begin{cases} \bar{r}_i + 1 & \text{w.p. } p^{\bar{a}_i}, \\ \bar{r}_i - 1 & \text{w.p. } q^{\bar{a}_i}, \\ \bar{r}_i & \text{w.p. } 1 - p^{\bar{a}_i} - q^{\bar{a}_i}. \end{cases} \tag{5}$$

For a state action pair $((\bar{a}, \bar{r}), a)$, the expected single state cost is given by $c((\bar{a}, \bar{r}), a)$ defined in the previous section (see (3)). A policy π is a sequence of mappings $\{u_t^\pi, t = 1, 2, \cdots\}$ where $u_t^\pi : \mathcal{A} \times \mathbb{Z}_+^K \to \mathcal{A}$. The cost of a policy π for an initial state (r, a) is

$$V^\pi(a, r) := \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^t c(x(t), u_t^\pi(x(t))) \middle| x(0) = (a, r)\right]$$

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

[2] One can as well consider minimizing expected value of $\sum_{t=1}^{\infty} \sum_{i \in \mathcal{C}} \beta_i^t c_i(a_i(t-1), r_i(t-1), a_i(t))$. This would model the scenario where the contents have geometrically distributed lifetimes with parameters β_is. Our RMAB-based solution continues to apply in this case.

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 [4, Vol 2, Chap. 1], the content caching problem assumes an optimal stationary policy.

Restless multi-armed bandit formulation The Markov decision problem described above presents a challenge due to its high dimensionality. However, we can make the following observations:

1. The evolution of the popularities of the contents is independent when considering their caching statuses. Their popularity changes are connected solely through the caching actions, specifically the capacity constraint of the cache.
2. The total cost can be divided into individual costs associated with each content.

We thus see that the optimal caching problem is an instance of the *restless multi-armed bandit problem* (RMAB) with each arm representing content. We show in Sect. 3 that this problem is *indexable*. This allows us to develop a Whittle index policy for the joint caching problem.

By recognizing the similarities between the optimal caching problem and the *restless multi-armed bandit problem* (RMAB), where each arm corresponds to a content, we can conclude that the optimal caching problem can be framed as an RMAB instance. In Sect. 3, we establish that this problem is *indexable*, enabling us to devise a Whittle index policy to tackle the joint caching problem efficiently.

Remark 1. In practice the popularity evolution could be unknown. The authors in [14,21–23] have proposed reinforcement learning (RL) based approaches to learn Whittle indices in case of Markov chains with unknown dynamics. But these works consider only one-dimensional Markov chains. Thanks to the switching costs, we have a two-dimensional Markov chain at our disposal which renders convergence of the RL algorithms and calculation of the Whittle indices much harder. This constitutes our future work.

3 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. Under indexability, the solution to the caching problem corresponding to a content yields Whittle indices for all the states of this content. The Whittle index measures how rewarding it is to cache that content at that particular state. The Whittle index policy chooses those M arms whose current states have the largest Whittle indices and, among these caches, those with positive Whittle indices.

3.1 Single Content Caching Problem

We consider a Markov decision problem associated with Content i. Its state space is $\{0, 1\} \times \mathbb{Z}_+$ and its action space is $\{0, 1\}$. Given $x_i(t) = (\bar{a}_i, \bar{r}_i)$ and action

$a_i(t) = a_i$, the state evolves as described in Sect. 2 (see (5)). The expected single-stage cost is

$$c_{i,\lambda}(\bar{a}_i, \bar{r}_i, a_i) = \lambda a_i + d a_i(1 - \bar{a}_i) + (1 - a_i) T^{a_i} C_i(r_i). \tag{6}$$

Observe that a constant penalty λ is incurred in each slot in which Content i is stored in the cache. Here a policy π is a sequence $\{u_t^\pi, t = 1, 2, \cdots\}$ where $u_t^\pi : \{0,1\} \times \mathbb{Z}_+ \to \{0,1\}$. Given initial state (a_i, r_i) the problem minimizes

$$V_{i,\lambda}^\pi(a_i, r_i) := \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^t c_{i,\lambda}(x_i(t), u_t^\pi(x_i(t))) \Big| x_i(0) = (a_i, r_i)\right]$$

over all the policies to yield the optimal cost function $V_\lambda(a_i, r_i) := \min_\pi V_{i,\lambda}^\pi(a_i, r_i)$. We analyze this problem below. However, we omit the index i for brevity. The single content caching problem also assumes an optimal stationary policy. Moreover, following [4, Vol 2, Chap. 1], the optimal cost function $V_\lambda(\cdot, \cdot)$ satisfies the following Bellman's equation.

$$V_\lambda(a, r) = \min_{a' \in \{0,1\}} Q_\lambda(a, r, a'), \tag{7}$$

where

$$Q_\lambda(a, r, a') := c_\lambda(a, r, a') + \beta T^{a'} V_\lambda(a', r). \tag{8}$$

Here $T^{a'} V_\lambda(a', r)$ is defined by applying operator $T^{a'}$ to the function $V_\lambda(a', \cdot) : \mathbb{Z}_+ \to \mathbb{R}_+$. The set $\mathcal{P}(\lambda)$ of the states in which action 0 is optimal, referred to as the *passive set*, is given by

$$\mathcal{P}(\lambda) = \{(a, r) : Q_\lambda(a, r, 0) \leq Q_\lambda(a, r, 1)\}.$$

The complement of $\mathcal{P}(\lambda)$ is referred to as the *active set*. Obviously, the penalty λ influences the partition of the passive and active sets.

Definition 2. (*Indexability*) An arm is called indexable if the passive set $\mathcal{P}(\lambda)$ of the corresponding content caching problem monotonically increases from $\emptyset$ to the whole state space $\{0,1\} \times \mathbb{Z}_+$ as the penalty λ increases from $-\infty$ to ∞. An RMAB is called indexable if it's every arm is indexable [3].

The minimum penalty needed to move a state from the active set to the passive set measures how attractive this state is. This motivates the following definition of the Whittle index.

Definition 3. (*Whittle index*) If an arm is indexable, its Whittle index $w(a, r)$ associated with state (a, r) is the minimum penalty that moves this state from the active set to the passive set. Equivalently,

$$w(a, r) = \min\{\lambda : (a, r) \in \mathcal{P}(\lambda)\}. \tag{9}$$

Before we establish the indexability of the arm (content) under consideration, we define threshold policies and show that the optimal policy for the single content caching problem is a threshold policy.

Definition 4. (*Threshold Policies*) A stationary policy u is called a threshold policy if it is of the form

$$u(a,r) = \begin{cases} 0 \text{ if } r \leq r^a, \\ 1 \text{ otherwise,} \end{cases}$$

for some $r^a \in \mathbb{Z}_+, a = 0, 1$. In the following, we refer to such a policy as the threshold policy (r^0, r^1).

3.2 Optimality of a Threshold Policy

Observe that the optimal cost function $V_\lambda(\cdot,\cdot)$ is obtained as the limit of the following *value iteration* [4, Vol 2, Chap. 1]. For all $(a,r) \in \{0,1\}\times\mathbb{Z}_+$, $V_\lambda^0(a,r) = 0$, and for $n \geq 1$,

$$V_\lambda^n(a,r) = \min_{a' \in \{0,1\}} Q_\lambda^n(a,r,a'), \tag{10}$$

where

$$Q_\lambda^n(a,r,a') := c_\lambda(a,r,a') + \beta T^{a'} V_\lambda^{n-1}(a',r). \tag{11}$$

We start by arguing that $V_\lambda^n(a,r)$ is concave and increasing in r for all a and $n \geq 0$. But it requires the following assumption.

Assumption 3.

$$p^0\Big(C(3)-C(2)\Big)-(2p^0+q^0-1)\Big(C(2)-C(1)\Big)+(p^0+2q^0-1)\Big(C(1)-C(0)\Big) \leq 0.$$

Lemma 1. *$V_\lambda^n(a,r)$ is concave and non-decreasing in r for all a and $n \geq 0$.*

Proof. Please refer to our extended version [24] □

Remark 2. We require Assumption 3 merely to prove that $V_\lambda^1(a,r)$ is concave. It follows via induction that $V_\lambda^n(a,r), n \geq 2$ are also concave.

Lemma 2. *For all $n \geq 1$,*

1. *$Q_\lambda^n(a,r,0) - Q_\lambda^n(a,r,1)$ are non-decreasing in r for $a = 0, 1$.*
2. *$V_\lambda^n(0,r) - V_\lambda^n(1,r)$ are non-decreasing in r.*

Proof. Please refer to our extended version [24] □

The following theorem uses Lemma 1 and 2 to establish that there exist optimal threshold policies for the single content caching problems.

Theorem 1. *For each $\lambda \in \mathbb{R}$ there exist $r^0(\lambda), r^1(\lambda) \in \mathbb{Z}_+$ such that the threshold policy $(r^0(\lambda), r^1(\lambda))$ is an optimal policy for the single content caching problem with penalty λ. Also, $r^0(\lambda) \geq r^1(\lambda)$.*

Proof. Please refer to our extended version [24]. □

3.3 Indexability of the RMAB

We now exploit the existence of an optimal threshold policy to argue that the RMAB formulation of the content caching problem is indexable.

Lemma 3. *For all* $n \geq 1$,

1. $Q_\lambda^n(a, r, 1) - Q_\lambda^n(a, r, 0)$ *are non-decreasing in* λ *for* $a = 0, 1$.
2. $V_\lambda^n(1, r) - V_\lambda^n(0, r)$ *are non-decreasing in* λ.
3. $V_\lambda^n(a, r+1) - V_\lambda^n(a, r)$ *are non-decreasing in* λ *for* $a = 0, 1$.

Proof. Please refer to our extended version [24]. □

Theorem 2. *Under Assumptions 1, 2 and 3 the content caching problem is indexable.*

Proof. Please refer to our extended version [24]. □

Remark 3. A more common approach to show indexibility of a RMAB have been arguing that the value function is convex, e.g., see [7,25]. But it can be easily verified that the value functions in our problem will not be convex even if the content missing costs are assumed to be convex.

Remark 4. Theorems 1 and 2 require Assumption 3. Many works in literature have relied on such conditions on transition probabilities for indexibility of RMABs (see [26,27]).

3.4 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 M arms whose current states have the largest Whittle indices and among these caches the ones with positive Whittle indices. It is a stationary policy. Let $u^W : \mathcal{A} \times \mathbb{Z}_+^K \to \mathcal{A}$ denote this policy. Then

$$u_i^W(a, r) = \begin{cases} 1 \text{ if } w(a_i, r_i) \text{ is among the highest } M \text{ values} \\ \quad \text{in } \{w(a_i, r_i), i \in \mathcal{C}\} \text{ and } w(a_i, r_i) > 0, \\ 0 \text{ otherwise.} \end{cases}$$

Holding cost There can also be a holding cost for keeping a 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

$$c_i(\bar{a}_i, \bar{r}_i, a_i) = ha_i + da_i(1 - \bar{a}_i) + (1 - a_i)T^{a_i}C_i(r_i).$$

Comparing it with (6), we see that the penalty λ 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.

Table 1. Choices of $\hat{C}(0)$ and $\hat{C}(1)$ that render the problem indexable; empty cells indicate absence of suitable choices.

Case	Costs
$p^0 + 2q^0 \leq 1$	$\hat{C}(1) = C(1)$, $\hat{C}(0) = \min\left\{C(0), C(1) - \frac{F}{p^0+2q^0-1}\right\}$
$p^0 + 2q^0 > 1, 2p^0 + q^0 < 1$	
$p^0 + 2q^0 > 1, 2p^0 + q^0 \geq 1, q^0 \leq p^0$	$\hat{C}(1) = C(2) - \frac{p^0(C(3)-C(2))}{p^0-q^0}$, $\hat{C}(0) = 2\hat{C}(1) - C(2)$
$p^0 + 2q^0 > 1, 2p^0 + q^0 \geq 1, q^0 > p^0$	

where $F = p^0\Big((C(2) - C(1)) - (C(3) - C(2))\Big) - (1 - p^0 - q^0)\Big(C(2) - C(1)\Big)$.

4 Noncompliance with Assumption 3

We have so far assumed that the costs and the transition probabilities satisfy Assumption 3 to ensure indexability of the content caching problem. We now explore a novel approach of manoeuvring the content missing costs so that the modified content caching problem is "close" to original problem and is indexable. We obtain the Whittle index policy for the modified problem and use it for the original problem.

More specifically, let us consider a particular content for which Assumption 3 is not met. We investigate the possibility of tinkering only $C(0)$ and $C(1)$ to achieve indexability.[3] In other words, we consider content missing costs $\hat{C} : \mathbb{Z}_+ \to \mathbb{R}_+$ with $\hat{C}(r) = C(r), r \geq 2$ and other costs and popularity evolution also unchanged. We demonstrate that in certain special cases adequate choices of $\hat{C}(0)$ and $\hat{C}(1)$ render the modified problem indexable. Our findings are summarized in Lemma 4 and Table 1.

Lemma 4. *If (a)* $p^0 + 2q^0 \leq 1$ *or (b)* $p^0 + 2q^0 > 1, 2p^0 + q^0 > 1, q^0 \leq p^0$ *then, with* $\hat{C}(0)$ *and* $\hat{C}(1)$ *as in Table 1,*

1. $\hat{C} : \mathbb{Z}_+ \to \mathbb{R}_+$ *is non-decreasing,*
2. $\hat{C} : \mathbb{Z}_+ \to \mathbb{R}_+$ *is concave,*
3. *the modified costs satisfy Assumption 3.*

In other cases, there do not exist $\hat{C}(0)$ *and* $\hat{C}(1)$ *such that* $\hat{C} : \mathbb{Z}_+ \to \mathbb{R}_+$ *satisfies all these three properties.*

Proof. Please refer to our extended version [24]. □

Typically the number of requests remains much higher than 0 or 1, and so the optimal caching policy and the cost for the amended problem are close to those for the original problem. Consequently, the Whittle index policy for the modified problem also performs well for the original problem. We demonstrate it in Sect. 6 though theoretical performance bounds have eluded us so far.

[3] As in Sect. 3.1, we omit the content index.

5 Popularity Evolution Oblivious to the Caching Action

If the popularity evolution is independent of the caching action, $p^0 = p^1$ and $q^0 = q^1$. In this case one can easily show via induction that $V_\lambda(1,r) - V_\lambda(0,r)$ is non-decreasing in λ, and consequently, the associated restless bandit problem is indexable. The proofs of these assertions follow from similar steps as in the proofs of the second part of Lemma 3 and of Theorem 2, respectively.

Remark 5. When popularity evolution does not depend on the caching action the above indexability assertion does not rely on the the special transition structure of Fig. 1. In other words, the restless bandit problems corresponding to the caching problems with arbitrary Markovian popularity evolutions are indexable as long as the evolutions do not depend on the caching action.

Below we generalize the above assertion by deriving a condition on the difference of popularity evolution with and without caching for indexability. Towards this, we define $\delta := 2\max\{|p^0 - p^1|, |q^0 - q^1|\}$. We prove the following indexability result.

Theorem 3. *If $\beta \leq \max\{\frac{1}{1+\delta}, \frac{1}{2}\}$, the content caching problem is indexable.*

Proof. We argue that

1. $Q^1_\lambda(a,r,1) - Q^1_\lambda(a,r,0)$ is nondecreasing in λ,
2. If $Q^n_\lambda(a,r,1) - Q^n_\lambda(a,r,0)$ for $a = 0,1$ are nondecreasing in λ, so is $V^n_\lambda(1,r) - V^n_\lambda(0,r)$,
3. If $V^n_\lambda(1,r) - V^n_\lambda(0,r)$ is nondecreasing in λ and $\beta \leq \max\{\frac{1}{1+\delta}, \frac{1}{2}\}$, then $Q^{n+1}_\lambda(a,r,1) - Q^{n+1}_\lambda(a,r,0)$ is also nondecreasing in λ.

Hence, via induction, $Q_\lambda(a,r,1) - Q_\lambda(a,r,0)$ is nondecreasing in λ which implies that the problem is indexable. Please see [24] for the details. □

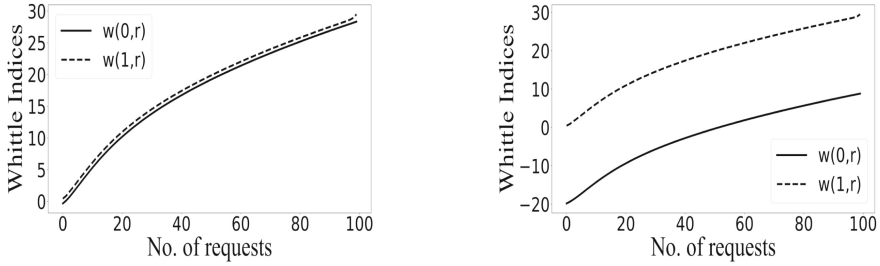
Observe that, for $\beta \leq 1/2$, the problem is indexable for arbitrary p^0, p^1, q^0 and q^1. For $\beta \in (1/2, 1)$, the problem is indexable if $\delta \leq \frac{1-\beta}{\beta}$. Assumptions 1, 2 or 3 are not needed in these cases.

Remark 6. Meshram et al. [27] have considered a restless single-armed hidden Markov bandit with two states and have shown that under certain conditions on transition probablities, the arm is indexable for $\beta \in (0, 1/3)$. More recently, Akbarzadeh and Mahajan [28] have provided sufficient conditions for indexability of general restless multiarmed bandit problems. One can directly infer from [28, Theorem 1 and Proposition 2] that the caching problem is indexable for $\beta \leq 1/2$. On the other hand, Theorem 3 implies that the caching problem can be indexable even for $\beta \in (1/2, 1)$. In Sect. 6, we numerically show that the caching problem can be indexable even when $\beta > 1/(1+\delta)$.

6 Numerical Results

In this section, we numerically evaluate the Whittle index policy for a range of system parameters. We demonstrate a variation of Whittle indices with various parameters. We also compare the performance of the Whittle index policy with those of the optimal and greedy policies. We compute Whittle indices using an algorithm proposed in [29]. We assume $C_i(r) = 3\sqrt{r}$ for all $i \in \mathcal{C}$. We use $p^{a_i}, q^{a_i}, a \in \{0, 1\}$, for all $i \in \mathcal{C}$ which satisfy Assumptions 1 and 3.

Whittle indices for different caching costs We consider $K = 40$ contents and a cache size $M = 16$. We assume transition probabilities $p^0 = 0.06082, q^0 = 0.38181, p^1 = 0.63253, q^1 = 0.26173$ for all the contents and discount factor $\beta = 0.95$. We plot the Whittle indices for two different values of the caching costs $d = 10$ and $d = 400$ in Figs. 2(a) and 2(b), respectively. As expected, $w(0, r)$ and $w(1, r)$ are increasing in r. We also observe that, for a fixed r, $w(1, r)$ does not vary much as the caching cost is changed but $w(0, r)$ decreases with the caching cost. This is expected as increasing caching costs makes the passive action (not caching) more attractive.



(a) Caching cost $d = 10$ (b) Caching cost $d = 400$

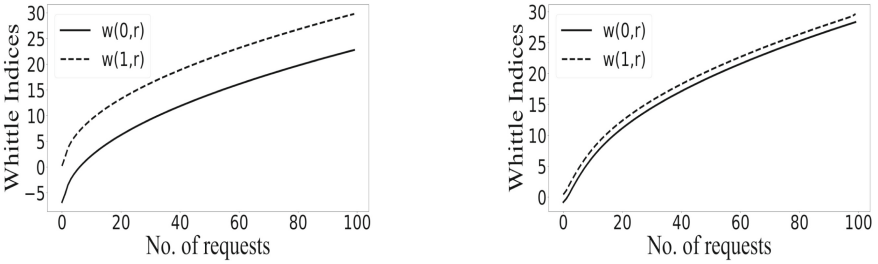
Fig. 2. Whittle indices for different caching costs

Whittle indices for different discount factors We plot the Whittle indices for two different values of discount factors $\beta = 0.3$ and $\beta = 0.9$ in Figs. 3(a) and 3(b), respectively. Other parameters are the same as those for Fig. 2 except caching cost $d = 10$. Here also, we observe that, for a fixed r, $w(1, r)$ does not vary much as the β is changed, but $w(0, r)$ increases with β.

Whittle indices for different transition probabilities We plot the Whittle indices for two different sets of transition probabilities

1. $p^1 > q^1$ and $p^0 > q^0$ e.g., $p^0 = 0.0093, q^0 = 0.0061, p^1 = 0.3274, q^1 = 0.0030$
2. $p^1 < q^1$ and $p^0 < q^0$ e.g., $p^0 = 0.0007, q^0 = 0.9362, p^1 = 0.0021, q^1 = 0.8902$

Assuming caching cost $d = 10$ and $\beta = 0.95$, we note that in the first scenario, whether the content is cached or not (active or passive action), the number of requests is likely to increase. Similarly, in the second scenario, the number of



(a) $\beta = 0.3$ (b) $\beta = 0.9$

Fig. 3. Whittle indices for different discount factors

requests is likely to decrease regardless of caching status. Figure 4 illustrates the Whittle indices for both cases. It is evident that the Whittle indices are higher in the first case compared to the second case. This outcome is expected since when the number of requests is more likely to increase, caching is anticipated. The Whittle index policy selects the M contents with the highest current state Whittle indices and caches those with positive indices. Thus, as the likelihood of request increases for content to be cached, the corresponding Whittle indices are expected to rise.

Performance of the Whittle index policy for a problem conforming to Assumptions 1 and 3 In our comparison, we evaluate the performance of the Whittle Index Policy against the optimal policy obtained through value iteration and the greedy policy. The greedy policy selects the action that minimizes the total cost outlined in Sect. 2, considering all possible actions while satisfying the constraints at each moment. However, the optimal policy is only feasible for small values of K, M. Therefore, we set $K = 3$ and $M = 1$ for our analysis. The parameters $\beta = 0.95$, caching cost $d = 10$, and the probabilities $p^{a_i}, q^{a_i}, a \in \{0, 1\}$, for all $i \in \mathcal{C}$, are chosen to satisfy Assumptions 1 and 3. Our findings indicate that the Whittle index policy outperforms the greedy policy and approaches the performance of the optimal policy, as depicted in Fig. 5.

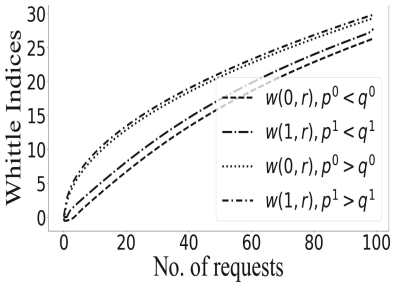


Fig. 4. Whittle indices for different transition probabilities

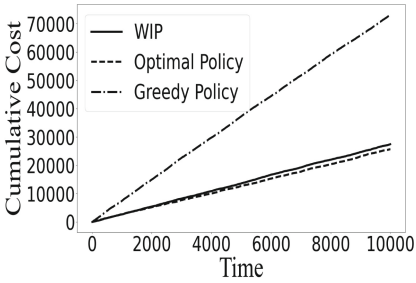


Fig. 5. Performance of Whittle Index Policy for a problem conforming to Assumptions 1 and 3

Performance of the Whittle index policy for a problem not conforming to Assumption 3 We manipulate the content missing costs of such a problem as suggested in Sect. 4 to make it indexable. In other words, we choose $\hat{C}(1)$ and $\hat{C}(0)$ as in the Table 1. We obtain the Whittle indices of the modified problem and use them for the original problem. Then we compare the performances of the Whittle index policy, the optimal policy and the greedy policy for the original problem. We see that the Whittle index policy still performs better compared to the greedy policy and is close to the optimal policy and is shown in Fig. 6

Indexability of the caching problem when $\delta > \frac{1-\beta}{\beta}$ The caching problem 4 is seen to be indexable when $\delta \leq \frac{1-\beta}{\beta}$. Here we numerically check indexability when $\delta > \frac{1-\beta}{\beta}$. We use $p^0 = 0.1855, p^1 = 0.2137, q^0 = 0.7719, q^1 = 0.6280$ and $\beta = 0.95$, resulting in $\delta = 0.2878$ and $\frac{1-\beta}{\beta} = 0.0526$. We run value iteration for different values of λ to find $Q_\lambda(a, r, a') \quad \forall a', a, r$ and plot $Q_\lambda(a, r, 1) - Q_\lambda(a, r, 0)$ vs λ. For indexability $Q_\lambda(a, r, 1) - Q_\lambda(a, r, 0)$ should be non-decreasing in λ as argued in the proof of Theorem 2. From Fig. 7, we can see that $Q_\lambda(a, r, 1) - Q_\lambda(a, r, 0)$ is indeed non-decreasing in λ. Hence the caching problem is indexable even when $\delta > \frac{1-\beta}{\beta}$.

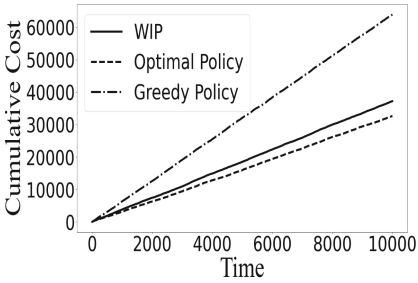


Fig. 6. Performance of Whittle Index Policy for a problem not conforming to Assumption 3

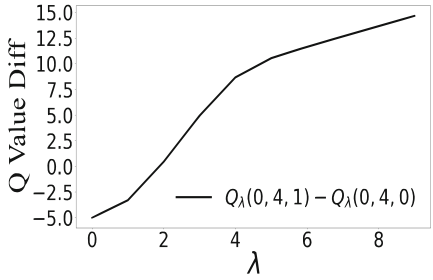


Fig. 7. Indexability of caching problem when $\delta > \frac{1-\beta}{\beta}$.

7 Conclusion

We considered optimal caching of contents with varying popularity. We posed the problem as a discounted cost Markov decision problem and showed that it is a an instance of RMAB. We provided a condition under which its arms is indexable and also demonstarted the performance of the Whittle index policy.

Our future work entails deriving performance bounds of the Whittle index policy. We plan to consider more general popularity dynamics. We also aspire to augment RMAB model with reinforcement learning to deal with the scenarios where the popularity dynamics might be unknown.

References

1. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf
2. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley (1994)
3. Whittle, P.: Restless bandits: activity allocation in a changing world. J. Appl. Prob. **25**(A), 287–298 (1988)
4. Bertsekas, D.: Dynamic Programming and Optimal Control, I and II, Athena Scientific, Belmont, Massachusetts. New York-San Francisco-London (1995)
5. Glazebrook, K., Mitchell, H.: An index policy for a stochastic scheduling model with improving/deteriorating jobs. Naval Res. Logistics (NRL) **49**(7), 706–721 (2002)
6. Glazebrook, K.D., Ruiz-Hernandez, D., Kirkbride, C.: Some indexable families of restless bandit problems. Adv. Appl. Probab. **38**(3), 643–672 (2006)
7. Ansell, P., Glazebrook, K.D., Nino-Mora, J., O'Keeffe, M.: Whittle's index policy for a multi-class queueing system with convex holding costs. Math. Methods Oper. Res. **57**(1), 21–39 (2003)
8. Li, S., Xu, J., Van Der Schaar, M., Li, W.: Popularity-driven content caching. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, pp. 1–9 (2016)
9. Sadeghi, A., Sheikholeslami, F., Giannakis, G.B.: Optimal and scalable caching for 5G using reinforcement learning of space-time popularities. IEEE J. Selected Topics Signal Process. **12**(1), 180–190 (2017)

10. Gao, J., Zhang, S., Zhao, L., Shen, X.: The design of dynamic probabilistic caching with time-varying content popularity. IEEE Trans. Mob. Comput. **20**(4), 1672–1684 (2020)
11. Abani, N., Braun, T., Gerla, M.: Proactive caching with mobility prediction under uncertainty in information-centric networks. In: Proceedings of the 4th ACM Conference on Information-Centric Networking, pp. 88–97 (2017)
12. Traverso, S., Ahmed, M., Garetto, M., Giaccone, P., Leonardi, E., Niccolini, S.: Temporal locality in today's content caching: why it matters and how to model it. ACM SIGCOMM Comput. Commun. Rev. **43**(5), 5–12 (2013)
13. ElAzzouni, S., Wu, F., Shroff, N.,Ekici, E.: Predictive caching at the wireless edge using near-zero caches. In: Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, pp. 121–130 (2020)
14. Xiong, G., Wang, S., Li, J., Singh, R.: Model-free reinforcement learning for content caching at the wireless edge via restless bandits (2022). arXiv preprint arXiv:2202.13187
15. Le Ny, J., Feron, E.: Restless bandits with switching costs: linear programming relaxations, performance bounds and limited lookahead policies. In: 2006 American Control Conference, p. 6 (2006)
16. Aalto, S., Lassila, P., Osti, P.: Whittle index approach to size-aware scheduling with time-varying channels. In: Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 57–69 (2015)
17. Anand, A., de Veciana, G.: A whittle's index based approach for QoE optimization in wireless networks. In: Proceedings of the ACM on Measurement and Analysis of Computing Systems **2**(1), 1–39 (2018)
18. Duran, S., Ayesta, U., Verloop, I.M.: On the whittle index of markov modulated restless bandits. Queueing Syst. (2022)
19. Sadeghi, A., Wang, G., Giannakis, G.B.: Deep reinforcement learning for adaptive caching in hierarchical content delivery networks. IEEE Trans. Cognit. Commun. Netw. **5**(4), 1024–1033 (2019)
20. Wu, P., Li, J., Shi, L., Ding, M., Cai, K., Yang, F.: Dynamic content update for wireless edge caching via deep reinforcement learning. IEEE Commun. Lett. **23**(10), 1773–1777 (2019)
21. Avrachenkov, K.E., Borkar, V.S.: Whittle index based Q-learning for restless bandits with average reward. Automatica **139**, 110186 (2022)
22. Fu, J., Nazarathy, Y., Moka, S., Taylor, P.G.: Towards Q-learning the whittle index for restless bandits. In: Australian & New Zealand Control Conference (ANZCC). IEEE, vol. 2019, pp. 249–254 (2019)
23. Robledo, F., Borkar, V., Ayesta, U., Avrachenkov, K.: QWI: Q-learning with whittle index. ACM SIGMETRICS Performance Eval. Rev. **49**(2), 47–50 (2022)
24. Pavamana, K.J., Singh, C.: Caching contents with varying popularity using restless bandits (2023). https://arxiv.org/pdf/2304.12227.pdf
25. Larranaga, M., Ayesta, U., Verloop, I.M.: Index policies for a multi-class queue with convex holding cost and abandonments. In: The ACM International Conference on Measurement and Modeling of Computer Systems **2014**, 125–137 (2014)
26. Liu, K., Zhao, Q.: Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. IEEE Trans. Inf. Theory **56**(11), 5547–5567 (2010)

27. Meshram, R., Manjunath, D., Gopalan, A.: On the whittle index for restless multiarmed hidden Markov bandits. IEEE Trans. Autom. Control **63**(9), 3046–3053 (2018)
28. Akbarzadeh, N., Mahajan, A.: Conditions for indexability of restless bandits and an $\mathcal{O}(k^3)$ algorithm to compute whittle index. Adv. Appl. Probab. **54**(4), 1164–1192 (2022)
29. Gast, N., Gaujal, B., Khun, K.: Testing indexability and computing whittle and gittins index in subcubic time. Math. Methods Oper. Res. **97**(3), 391–436 (2023). https://doi.org/10.1007/s00186-023-00821-4