Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

UNIVERSAL WISER PUBLISHER

Article

Scheduling Service Updates: A Multi-arm Bandit Approach

V S Ch Lakshmi Narayana[1], Sucheta Ravikanti[2], Harsh Deshpande[3] and Sharayu Moharir[4,*]

[1] Modem Systems Engineer, Qualcomm India Pvt Ltd, Bengaluru, 560066, India
[2] Fixed Income Division Associate, Morgan Stanley, Mumbai, 400098, India
[3] Department of Computer Science and Engineering, University of California San Diego, CA 92093, USA
[4] Department of Electrical Engineering, Indian Institute of Technology, Bombay, 400076, India
E-mail: sharayu.moharir@gmail.com

Received: 29 July 2022; **Revised:** 24 November 2022; **Accepted:** 25 November 2022

Abstract: Software as a Service (SaaS) instances often use edge resources to serve their customers. The version of the service hosted at the edge needs to be periodically updated to maximize the utility derived by the customers. We focus on scheduling updates in the setting where the utility derived from a version is an unknown decreasing function of the time elapsed since the version was created. We map the scheduling problem to a multi-arm bandit and propose an update policy. We characterize its performance and compare it with the fundamental limit on the performance of any online policy.

Keywords: edge computing, multi-arm bandits, scheduling

1. Introduction

Many real-time applications/services like weather forecasting, online shopping, and GPS-based navigation need fresh data and the latest software to provide maximum benefit to their users. For example, online shopping websites typically have an evolving catalogue of items for sale. In addition, the prices of the various items for sale also evolve over time. Another example is when the code of a GPS-based navigation application is updated to fix existing bugs or add new features. Many such services are hosted at the edge, i.e., either on servers close to the end-users or the users' local devices. Typically, the latest version of the service is more useful to the user than older versions. Therefore, the version of the service hosted at the edge needs to be updated recurrently to ensure good quality of service to the users. Each such service update requires new data to be brought to the edge via the Internet and therefore leads to bandwidth consumption.

Motivated by this trade-off between quality of service and bandwidth consumption, we consider a time-slotted system and model the reward the user derives from the service currently hosted at the edge as a decreasing function of the time elapsed since the service hosted at the edge was last updated. Also, the amount of bandwidth consumed by an update is modelled as an increasing function of the time elapsed since the last update. This is because a longer inter-update time results in a larger difference in the latest iteration and the currently hosted iteration of the service and therefore, more data needs to be fetched to update the service to the latest version on the edge device. The utility in a time-slot is defined as the difference between the reward gained and the update cost incurred in the time-slot. This is motivated by the fact that typically with time, service developers add more features to the service, thus increasing the size of the source code and other auxiliary files. Since an update would require us to fetch these files to the edge, larger the size of the source code and other auxiliary files, higher the bandwidth consumption.

The algorithmic task is to schedule service up-dates in the setting where the reward derived by the user from the currently hosted version of the service is an unknown function of the time elapsed since the recent most

DOI: https://doi.org/10.37256/cnc.1120231686

update and needs to be estimated through user feedback. The goal is to design a policy to maximize utility. We map our setting to a multi-arm bandit (MAB) and propose a policy (a variant of UCB [1]) that determines when to update the service at the edge. We also characterize the fundamental limit on the performance of any policy in this setting and use that as the benchmark to evaluate the performance of the proposed policy. Our analytical results are not trivial extensions of known results in MAB literature since the problem has a specific structure which we carefully exploit to prove our results.

1.1 *Related Work*

Our work has connections to the body of work on the Age-of-Information (AoI) metric. Closest to our setting, [2–5] focus on caching policies that determine which contents to cache in order to minimize specific functions of the AoI of the requested content. In [2] the goal is to update the cache to minimize the expected AoI of the requested file given the popularity of the files in the cata-log. The problem is formulated as an optimization problem. A relaxed version of this optimization problem is solved to design a practical cache update policy. In [3], the goal is to design a dynamic cache update schedule when the performance metric is a decreasing function of AoI. Guarantees on problem tractability are provided and a scalable solution approach is proposed. On similar lines, [4] proposes a user's queue-aware cache update scheduling algorithm while focusing on minimizing the average AoI. In this work, the problem is formulated as a Markov Decision Process. The key difference between these works and our setting is that in [2–4] the utility of each cached content is a *known function* of its AoI. In [5], the goal is to design a cache update scheme for Internet of Things (IoT) networks. The authors propose an online cache update scheme to obtain a trade-off between average AoI and the energy consumed by the IoT sensors. This work does not assume any knowledge on user preferences towards contents and uses deep reinforcement learning to design a cache update scheme. Specifically, this work uses Deep Q-Network (DQN) and the parameters of Q-value functions in DQN are updated using a gradient descent algorithm.

In [6–10], the focus is on designing scheduling policies to minimize AoI for one or more sources sending updates to a monitoring station via multiple communication channels in the setting where channel statistics are unknown. This body of work also maps the scheduling problem to the multi-arm bandit problem. Unlike these works, our goal is to maximize utility which is an *unknown function* of the AoI. The cost incurred on each update in our setting is also a key difference between [6–10] and our setting.

We map our problem to the MAB problem with a specific structure. Other variants of the MAB with structure that have been studied include [11–17] where the rewards of arms are correlated through a common hidden parameter.

2. Setting

2.1 *System Model*

We consider a service installed on an edge device. Time is divided into slots. In each time-slot, a new version of the service is released by the developer. On request, the version of the service currently hosted on the edge device is replaced by the latest version of the service. We refer to this as an update.

Definition 1 (Age of Installed Version). *The age of the version installed on the edge device in time-slot t is denoted by $a(t)$ and is a measure of the time-elapsed since the version currently installed on the edge device was released by the service developer. Formally,*

$$a(t) = \begin{cases} 1 & \textit{if an update requested in time-slot } t \\ a(t-1)+1 & \textit{otherwise.} \end{cases}$$

Note that we assume that if the edge device requests for a version update in time-slot t, the update is delivered to it by the end of time-slot t.

A cost is incurred on each update and is an increasing function of the time elapsed since the previous update. Let c_i denote the cost incurred for an update requested after a gap of i time-slots. The formal definition of the cost incurred in a time-slot is as follows.

Definition 2 (Cost). *Let $c(t)$ denote the cost incurred in time-slot t. It follows that*

$$c(t) = \begin{cases} c_{a(t\text{-}1)} & \textit{if an update requested in time-slot } t \\ 0 & \textit{otherwise.} \end{cases}$$

Assumption 1. $c_i \le c_j$ *for* $i < j$.

The reward derived by the user from the service is a decreasing function of the age of the installed version. The formal definition of reward is as follows.

Definition 3 (Reward). *Let* $r(t)$ *denote the reward accrued in time-slot t. We consider the setting where* $r(t) \in \{0, 1\}$ *with*

$$r(t) = \begin{cases} 1 & w.p.\ \mu_{a(t)} \\ 0 & \textit{otherwise,} \end{cases}$$

and $\mathrm{E}\left[r(t)\right] = \mu_{a(t)}$.

This Bernoulli reward assumption can be mapped to binary feedback from the user of the service, i.e., the reward accrued in time-slot t is $r(t) = 1$, if the user is satisfied with the version of the service hosted at the edge in a time-slot and $r(t) = 0$ otherwise.

Assumption 2. $\mu_i \ge \mu_j$ *for* $i < j$.

The costs are assumed to be known since there is a measure of the difference between two versions of the service which the service developer can design/predict whereas the rewards depend on the user experience, which are often subjective and can only be inferred via user feedback.

We define the utility as a difference between the reward accrued and the cost incurred.

Definition 4 (Utility). *Let* u_t *denote the reward accrued in time-slot* t. *It follows that* $u_t = r(t) - c(t)$.

2.2 *Performance Metric*

We consider the setting where the values of the expected rewards, i.e., the $\mu_i s$ are unknown. The cost values, i.e., the $c_i s$ are known. Further, we impose an upper limit of M time-slots on the time between two consecutive updates.

Definition 5 (Optimal Inter-Update Period). *For each inter-update period* $i \in \{1, 2, ..., M\}$, *we define its score as the infinite horizon* ($T \to \infty$) *time-average of expected utility when the service is updated every i time-slots. The optimal inter-update period* ($i^* \in \{1, 2, ..., M\}$) *is defined as the inter-update period with the maximum score.*

Let $u^{\mathrm{P}}(t)$ be the utility in time-slot t under a given policy P, and let $u^*(t)$ be the utility in time-slot t under the genie policy that always requests update at the optimum inter-update period (characterized in the next section). We define the utility regret at time T as the difference in the cumulative expected utility under the two policies in time-slots 1 to T.

Definition 6 (Utility Regret). *Utility regret under policy* P *is denoted by* $\Re^{\mathrm{P}}(T)$ *and* $\Re^{\mathrm{P}}(T) = \sum_{t=1}^{T} \mathrm{E}\left[u^*(t) - u^{\mathrm{P}}(t)\right]$.

2.3 *Initial Conditions*

We assume that at the beginning of the first time-slot, the service is not hosted at the edge device. Since the utility derived from the service is not defined for this case, we add the initial condition that all candidate policies, including the optimal policy, request an update in this time-slot, i.e., the service is fetched to the edge device in the first time-slot.

Assumption 3 (Initial Conditions). *The system starts operating in time-slot* $t = 1$ *and an update is requested in the first time-slot.*

2.4 *Goal*

The goal is to design a policy/algorithm to determine the optimum update frequency to use to request updates to minimize Utility Regret (Definition 6).

3. Main Results and Discussion

In this section, we present our analytical results. The proofs and simulation results are discussed in Section 4.

3.1 *Optimal Inter-update Period*

Theorem 1. *For a problem instance with reward vector* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ *and the cost vector* $c = \{c_1, c_2, ..., c_M\}$, *the optimal inter-update period is given by*

$$i^* = \arg\max_{1 \le i \le M} \frac{(\sum_{k=1}^{i} \mu_k) - c_i}{i}.$$

Remark 1. *There can be a reward vector* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ *and a cost vector* $c = \{c_1, c_2, ..., c_M\}$ *that gives multiple optimal inter-update periods* i^*. *Our analytical results in this paper are meaningful when* i^* *is unique. However, our results can be extended to the case when* i^* *is not unique by suitably updating the definition of certain quantities.*

Assumption 4. *The optimal inter-update period* i^* *described in Theorem* 1 *is unique.*

Remark 2. *Note that for a reward vector* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ *and a cost vector* $c = \{c_1, c_2, ..., c_M\}$, *if* $2c_i < c_j$ *for all* $i < j$ *then the optimal arm* $i^* = 1$.

The Genie policy which knows the values of the μis updates the service periodically with the optimal inter-update period characterized in Theorem 1.

3.2 *Our Policy: Utility-UCB*

We map our problem to a multi-arm bandit where the task of scheduling service updates is equivalent to the task of choosing one of M arms of the multi-arm bandit [18]. Time is partitioned into rounds with one arm chosen per round. Choosing arm m in a round is equivalent to not updating the service in the first $m-1$ time-slots of the round and updating the service in the m^{th} time-slot of the round.

Since the inter-update time belongs to the set $\{1, 2, ..., M\}$, a round lasts for up to M time-slots. Note that there is exactly one service update in each round and each round begins in the time-slot after an update is requested and ends when the next update is requested.

Next, we discuss our policy called Utility-UCB (U-UCB). This policy is a variant of the UCB policy which is known to be order-optimal for the MAB problem. Under the U-UCB policy, we maintain confidence bounds for the rewards ($\mu_m s, 1 \le m \le M$). The key difference between the classical UCB and our variant is the way in which these bounds are calculated. In Step 5 of the U-UCB policy (Algorithm 1), we use the empirical estimates of reward vector $\hat{\mu}_m s$ to compute $\hat{\gamma}_m$ for $1 \le m \le M$. Suppose we play arm $j \le M$ and therefore, observe samples of μ_i for $1 \le i \le j$. As a result, the estimates $\hat{\mu}_i$ for $1 \le i \le j$ get updated. As a consequence of this, $\hat{\gamma}_m$ for $1 \le m \le M$ are all updated. It, therefore, follows that samples of the unknown quantities ($\mu_i^{'} s$) obtained while playing one arm lead to an update in the upper confidence bounds of all other arms. We incorporate this fact that samples of the unknown quantities ($\mu_i s$) obtained while playing one arm leads to an update in the upper confidence bounds of all other arms. The policy starts with requesting an update in the first time-slot of round 1 consistent with Assumption 3. In the second round, the policy selects arm M, i.e., it requests an update in the M^{th} time-slot of the round. At the end of the first two rounds, we thus have at least one sample of each $\mu_m, 1 \le m \le M$. After the first two rounds, in each round, the arm/inter-update time is chosen as the value of j for which the time-averaged utility in the round using the upper confidence bounds of the μ_ms (computed in Step 5 of Algorithm 1) is maximized. Refer to Algorithm 1 for a formal definition.

Note that in Algorithm 1, r denotes the time index and the algorithm can potentially go on forever. Our performance guarantees hold for all (finite) time-horizons. We now characterize the performance of the U-UCB policy.

```
Algorithm1: Utility-UCB(U-UCB)
Input: $c_m$ for $1 \le m \le M$
Initialise: Set $\hat{\mu}_m = 0$ and $n_m = 0$
  $\forall m \in [M], r = 1, t = 1$
While $r > 0$ do
    for $m \in [M]$ do
        $\hat{\gamma}_m = \frac{\sum_{j=1}^{m} \hat{\mu}_j - c_m}{m} + \sqrt{\frac{2\log t}{m \times n_m}}$
      Choose arm $j$, where
        $j = \begin{cases} 1 & \text{if r} = 1 \\ M & \text{if r} = 2 \\ \arg\max_{m\in[M]} \hat{\gamma}_m & \text{otherwise}. \end{cases}$
      for $k \in [j]$ do
          Receive reward $X_k \sim \text{Ber}(\mu_k)$
          $\hat{\mu}_k = (\hat{\mu}_k \cdot n_k + X_k)/(n_k + 1)$
          $n_k = n_k + 1$
  $r = r + 1,\ t = t + j$
```

Theorem 2. *For a problem instance with reward vector* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ *and the cost vector* $c = \{c_1, c_2, ..., c_M\}$ *satisfying Assumption 4, the expected utility regret under U-UCB policy satisfies*

$$\mathrm{E}[\Re^{U\text{-}UCB}(T)] \le \frac{8M\log T}{\Delta_{min}}.$$

where

$$\Delta_{min} = \min_{m\in[M], m\neq i^*} \frac{\sum_{j=1}^{i^*} \mu_j - c_{i^*}}{i^*} - \frac{\sum_{j=1}^{m} \mu_j - c_m}{m}$$

for i^* *characterized in Theorem 1.*

We see that the upper bound on the utility regret under U-UCB policy is $O(\frac{M}{\Delta_{min}}\log T)$. The proof of Theorem 2 is discussed in Section 4.

3.3 *Lower Bound on Utility Regret*

The next theorem characterizes the lower bound on the utility regret under a special class of policies with certain restrictions on the optimal-inter update period and the structure of costs of updates ($c_i s$).

Assumption 5. *Let the reward vector be* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$, *the cost vector be* $c = \{c_1, c_2, ..., c_M\}$, *and* i^* *be the optimal inter-update period. Define* $c_0 = 0$. *Then,*

$$i^* \neq M \tag{1}$$

$$c_{i^*} \geq \frac{c_{i^*-1} + c_{i^*+1}}{2} \qquad (2)$$

Since our problem is a learning problem, it is important to characterize the fundamental limit on the performance of any online policy and we do this for a special class of parameter values in Theorem 3.

Theorem 3. *Consider a problem instance with re-ward vector* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ *and cost vector* $c = \{c_1, c_2, ..., c_M\}$ *satisfying Assumptions* 4 *and* 5. *Let* $\tilde{\mu}_j = \mu_j$ *for* $i \neq i^*$ *and* $\tilde{\mu}_{i^*} = \mu_{i+1}$ *and* $\tilde{i}^*$ *be the optimal inter-update period for the problem instance with reward vector* $\tilde{\mu} = \{\tilde{\mu}_1, \tilde{\mu}_2, ..., \tilde{\mu}_m\}$ *and cost vector* $c = \{c_1, c_2, ..., c_M\}$. *Let*

$$\Delta_m = \frac{\sum_{j=1}^{i^*} \mu_j - c_{i^*}}{i^*} - \frac{\sum_{j=1}^{m} \mu_j - c_m}{m}$$

$$\Delta_{min} = \min_{m \in [M], m \neq i^*} \Delta_m$$

$$\delta_m = \frac{\sum_{j=1}^{\tilde{i}^*} \tilde{\mu}_j - c_{\tilde{i}^*}}{\tilde{i}^*} - \frac{\sum_{j=1}^{m} \tilde{\mu}_j - c_m}{m}$$

$$\delta_{min} = \min_{m \in [M], m \neq \tilde{i}^*} \delta_m$$

and $D(\mu_j, \tilde{\mu}_j)$ *denote KL-divergence* [19] *between* μ_j *and* $\tilde{\mu}_j$. *Then, for any online policy P that achieves sub-polynomial utility regret,*

$$\mathrm{E}[\Re^{\mathrm{P}}(T)] \geq \frac{(i^* + 1)\Delta_{min}}{D(\mu_j, \tilde{\mu}_j)} \left[\log\left(\frac{\min\{\Delta_j, \delta_{min}\}}{8\zeta}\right) + (1-\beta)\log T \right]$$

For some $\zeta > 0$, *and* $0 < \beta < 1$.

The main challenge in proving this result comes from the fact that a valid instance for our problem has a specific structure. Specifically, the μ_is decrease with i and the c_is increase with i. In addition, pulling arm $k \in \{1, 2, ..., M\}$ gives one sample each of all arms indexed less than k. As a result of this structure of our problem, Theorem 3 does not follow from existing results for the MAB and requires novel arguments.

From Theorems 2 and 3, we conclude that under Assumption 5, U-UCB is order-optimal with respect to time. Order-optimality means that the upper bound on the regret of U-UCB and the lower bound on the regret of any online policy is proportional to log(*T*). Therefore, the rate at which the regret of U-UCB grows with time is as low as possible for any online policy. Further, for the special case when the optimal time-update period is $M - 1$, U-UCB is also order-optimal with respect to the upper limit on the inter-update time (*M*).

4. Proofs

In this section, we discuss the proofs of the results from Section 3.

4.1 *Proof of Theorem 1*

Proof. Let $i \in \{1, 2, ..., M\}$ be an inter-update period. Then by Definition 5, the score of i for the given problem instance is $\frac{(\sum_{k=1}^{i} \mu_k) - c_i}{i}$. Hence the optimal inter-update period is $i^* = \arg\max_{1 \leq i \leq M} \frac{(\sum_{k=1}^{i} \mu_k) - c_i}{i}$.

4.2 *Proof of Theorem 2*

We use the following lemma to prove Theorem 2.

Lemma 1. *For a problem instance with reward vector* $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ *and cost vector* $c = \{c_1, c_2, ..., c_M\}$ *satisfying Assumption* 4*, let* $n_i(t)$ *denote the number of times the inter-update time* i *will be used in the first* t *updates after* $t = 1$. *For all* $i \neq i^*$ (*optimal inter-update period*)*, let* $h_i(t) = \sum_{j=i}^{M} n_i(t)$ *and* $\Delta_i = \frac{\sum_{j=1}^{i^*} \mu_j - c_{i^*}}{i^*} - \frac{\sum_{j=1}^{m} \mu_j - c_i}{i}$ *then,*

$$h_i(t) < \frac{8 \log t}{i \cdot \Delta_i^2} \tag{3}$$

Proof. Let t be a time-slot in which a new round begins and $k(t)$ denote the inter-update time chosen by U-UCB policy in that round. It follows that

$$k(t) = \arg\max_{1 \le i \le M} \left(\frac{\tilde{g}_i - c_i}{i}\right) \tag{4}$$

Where $\tilde{g}_i = \sum_{j=1}^{i} (\hat{\mu}_j + \sqrt{\frac{2 \log t}{j \cdot h_j}})$. Note that the right-hand side of (4) is an increasing function of $\tilde{g}_i$. We define events A_t and B_t where $\forall j$,

$$A_t : \hat{\mu}_{j,n_j(t)} - \sqrt{\frac{2 \log t}{j \cdot h_j(t)}} \le \mu_j$$

$$B_t : \hat{\mu}_{j,n_j(t)} - \sqrt{\frac{2 \log t}{j \cdot h_j(t)}} \ge \mu_j$$

Here t, $n_j(t)$ and $\hat{\mu}_{j,n_j(t)}$ represent the time index, number of times arm j has been used by time-slot t, and the empirical estimate of μ_j after $n_j(t)$ pulls respectively. By [20], $\mathrm{P}(A_t^C)$, $\mathrm{P}(B_t^C) \le t^{-2}$. Conditioned on A_t, $\hat{b}_{j,n_j(t)} \le b_j + \sqrt{\frac{2 \log t}{j \cdot h_j(t)}}$, where $\hat{b}_{j,n_j(t)} = \hat{\mu}_{j,n_j(t)} + \sqrt{\frac{2 \log t}{j \cdot h_j(t)}}$ and $b_j = \mu_j + \sqrt{\frac{2 \log t}{j \cdot h_j(t)}}$. Using the increasing nature of the expression in (4),

$$\frac{\sum_{j=1}^{i} \hat{b}_{j,n_j(t)} - c_i}{i} \le \frac{\sum_{j=1}^{i} (b_j + \sqrt{\frac{2 \log t}{j \cdot h_j(t)}}) - c_i}{i} \tag{5}$$

A sub-optimal arm $i (\neq i^*)$ is pulled when the following is satisfied for all $p \in \{1, 2, ..., M\} \setminus i$,

$$\frac{\sum_{j=1}^{i} \hat{b}_{j,n_j(t)} - c_i}{i} \ge \frac{\sum_{j=1}^{p} \hat{b}_{j,n_j(t)} - c_p}{p} \tag{6}$$

Chaining (5) and (6),

$$\frac{\sum_{j=1}^{i} (b_j + \sqrt{\frac{2 \log t}{j \cdot h_j(t)}}) - c_i}{i} \ge \frac{\sum_{j=1}^{i^*} \hat{b}_{j,n_j(t)} - c_{i^*}}{i^*}$$

$$\text{i.e., } \frac{\sum_{j=1}^{i} 2 \sqrt{\frac{2 \log t}{j \cdot h_j(t)}})}{i} \ge \Delta_i$$

Now, $h_a \geq h_b$ if $a \leq b$, , i.e. $2\sqrt{\frac{2\log t}{j \cdot h_j(t)}}) \geq \Delta_i$, hence proving the lemma.

Proof of Theorem 2. Recall that $\Re^{\mathrm{P}}(T) = \sum_{i=1}^{M} i \cdot n_i \cdot \Delta_i$ We know that $n_i \leq h_i \; \forall i$. Therefore,

$$\Re^{\mathrm{P}}(T) \leq \sum_{i=1}^{M} i \cdot h_i \cdot \Delta_i \leq \sum_{i=1}^{M} \frac{8\log T}{\Delta_i} \leq \frac{8\log T}{\Delta_{\min}}.$$

4.3 *Proof of Theorem 3*

The main challenge comes from the fact that a valid instance for our problem has a specific structure. Specifically, the μ_is decrease with i and the c_is increase with i. In addition, pulling arm $k \in \{1, 2, ..., M\}$ gives one sample each of all arms indexed less than k. As a result of this structure of our problem, Theorem 3 does not follow from existing results for the MAB and requires novel arguments. We use the following lemma to prove Theorem 3.

Lemma 2. *Let I_1 be a bandit instance with the reward vector $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ and the cost vector $c = \{c_1, c_2, ..., c_M\}$ satisfying Assumption* 5. *Let $i^* \neq M$ be the optimal inter-update period under I_1. Consider an alternative instance I_2 with reward vector $\tilde{\mu} = \{\tilde{\mu}_1, \tilde{\mu}_2, ..., \tilde{\mu}_M\}$ and cost vector $\tilde{c} = c$. Let $\tilde{\mu}_{i^*+1} = \mu_{i^*}$ and $\tilde{\mu}_k = \mu_k$ for $k \in \{1, 2, ..., M\} \setminus (i^* + 1)$. Then, the optimal inter-update period under I_2 is $i^* + 1$.*

Proof. Let us prove this by contradiction, the index of the optimal arm in any bandit instance I_2 is less than or equal to i^*. From the Assumption 5 we have the following.

$$c_{i^*} \geq \frac{c_{i^*-1} + c_{i^*+1}}{2} \Rightarrow 2c_{i^*} \geq c_{i^*-1} + c_{i^*+1}.$$

Thus, we have $2i^* c_{i^*} \geq i^* c_{i^*-1} + i^* c_{i^*+1}$.

By rearranging the terms in the last inequality, we obtain the following,

$$(i^* + 1)c_{i^*} - i^* c_{i^*+1} \geq i^* c_{i^*-1} - (i^* + 1)c_{i^*}. \tag{7}$$

Since i^* is the optimal arm in I_1, we have the following.

$$\begin{gathered} \frac{\sum_{k=1}^{i^*} \mu_k - c_{i^*}}{i^*} > \frac{\sum_{k=1}^{i^*-1} \mu_k - c_{i^*-1}}{i^* - 1} \\ \frac{c_{i^*-1}}{i^* - 1} - \frac{c_{i^*}}{i^*} > \left(\frac{1}{i^* - 1} - \frac{1}{i^*}\right)\sum_{k=1}^{i^*-1} \mu_k - \frac{\mu_{i^*}}{i^*} \\ \Rightarrow i^* c_{i^*-1} - (i^* - 1)c_{i^*} > \sum_{k=1}^{i^*-1} \mu_k - (i^* - 1)\mu_{i^*} \end{gathered} \tag{8}$$

Consider a bandit instance I_2 with reward vector and cost vector same as I_1 except that $\mu_{i^*+1} = \mu_{i^*}$. That is the expected rewards in I_1 and I_2 differ only at the $i^* + 1$ position.
Assume that

$$\frac{\sum_{k=1}^{i^*} \mu_k - c_{i^*}}{i^*} > \frac{\sum_{k=1}^{i^*+1} \mu_k - c_{i^*+1}}{i^* - 1}$$

This implies,

$$\frac{c_{i^*}}{i^*} - \frac{c_{i^*+1}}{i^*+1} < (\frac{1}{i^*} - \frac{1}{i^*+1})\sum_{k=1}^{i^*-1}\mu_k + \frac{\mu_{i^*}}{i^*} - \frac{2\mu_{i^*}}{i^*+1},$$
$$(i^*+1)c_{i^*} - i^* c_{i^*+1} < \sum_{k=1}^{i^*-1}\mu_k - (i^*-1)\mu_{i^*} \tag{9}$$

This is a contradiction from (7) and (8).

Proof of Theorem 3. Let $\tilde{\mu}_k$ be the reward of any arm $k \in \{1, 2, ..., M\}$ under an alternative bandit instance I_2. From Lemma 2, there exists a bandit instance I_2 with optimal inter-update period $j > i^*$. Let $\eta_k(t)$ denote the total number of time-slots up to time-slot t corresponding to rounds in which an inter-update period $k \in \{1, 2, ..., M\}$ was chosen. The expected regrets after T rounds under the instance I_1 is,

$$\mathrm{E}_{I_1}[\Re(T)] = \sum_{k \neq i^*} \Delta_k \mathrm{E}_{I_1}[\eta_k(T)] \geq \Delta_j \mathrm{E}_{I_1}[\eta_j(T)]$$

and therefore,

$$\mathrm{E}_{I_1}[\eta_j(T)] \leq \frac{\mathrm{E}_{I_1}[\Re(T)]}{\Delta_j} \tag{10}$$

Similarly, the expected regrets after T rounds under instance I_2 is,

$$\mathrm{E}_{I_2}[\Re(T)] = \sum_{k \neq i^*} \delta_k \mathrm{E}_{I_2}[\eta_k(T)]$$

$$\sum_{k \neq i^*} \mathrm{E}_{I_2}[\eta_k(T)] \leq \frac{\mathrm{E}_{I_2}[\Re(T)]}{\delta_{\min}} \tag{11}$$

From the divergence decomposition theorem [18], we have

$$D(I_1, I_2) = \sum_{l=j}^{M} \frac{\mathrm{E}_{I_1}[\eta_l(T)]}{l} D(\mu_j, \hat{\mu}_j) \tag{12}$$

From the Bretagnolle–Huber inequality [18], we have for any event A

$$\mathrm{P}_{I_1}(A) + \mathrm{P}_{I_2}(A^C) \geq \frac{1}{2}\exp(-D(I_1, I_2)) \tag{13}$$

Let us consider the event $A := \{\eta_j(T) > \frac{T}{2}\}$ thus its complement $A^C := \{\sum_{k \neq j} \eta_k(T) > \frac{T}{2}\}$

Then by Markov inequality and (10) and (11) we have:

$$\mathrm{P}_{I_1}(A) \leq \frac{2}{T}\mathrm{E}_{I_1}[\eta_j(T)] \leq \frac{2\mathrm{E}_{I_1}[\Re(T)]}{T\Delta_j}$$

$$\mathrm{P}_{I_2}(A^C) \leq \frac{2}{T}\sum_{k \neq j}\mathrm{E}_{I_2}[\eta_k(T)] \leq \frac{2\mathrm{E}_{I_2}[\Re(T)]}{T\delta_{\min}}$$

Substituting in (13), we have

$$\frac{2\mathrm{E}_{I_1}[\Re(T)]}{T\Delta_j} + \frac{2\mathrm{E}_{I_2}[\Re(T)]}{T\delta_{\min}} \geq \frac{1}{2}\exp(-D(I_1, I_2))$$

For any policy that achieves sub-polynomial regret for all bandit instances, we have $\mathrm{E}_{I_1}[\Re(T)] + \mathrm{E}_{I_2}[\Re(T)] \leq 2\zeta T^{\beta}$ for $0 < \beta < 1$ and $\zeta > 0$. Thus,

$$\frac{4\zeta T^{\beta}}{\min\{\Delta_j,\delta_{\min}\}} \geq \frac{1}{2}\exp(-\sum_{l=j}^{M}\frac{\mathrm{E}_{I_1}[\eta_l(T)]}{l}D(\mu_j,\tilde{\mu}_j))$$

$$\Rightarrow \sum_{l=j}^{M}\frac{\mathrm{E}_{I_1}[\eta_l(T)]}{l} \geq \frac{1}{D(\mu_j,\tilde{\mu}_j)}\times[\log(\frac{\min\{\Delta_j,\delta_{\min}\}}{8\zeta})+(1-\beta)\log(T)]$$

$$\Rightarrow \frac{\mathrm{E}_{I_1}[\Re(T)]}{j\Delta_{\min}} \geq \frac{1}{D(\mu_j,\tilde{\mu}_j)}\times[\log(\frac{\min\{\Delta_j,\delta_{\min}\}}{8\zeta})+(1-\beta)\log(T)]$$

$$\Rightarrow \mathrm{E}_{I_1}[\Re(T)] \geq \frac{j\Delta_{\min}}{D(\mu_j,\tilde{\mu}_j)}\times[\log(\frac{\min\{\Delta_j,\delta_{\min}\}}{8\zeta})+(1-\beta)\log(T)]$$

We now compare the performance of our pro-posed policy (U-UCB (Algorithm 1)) with a variant of the Correlated UCB (C UCB) policy proposed in [21] via simulations. This variant called Utility-CUCB (U-CUCB) exploits the structure of the problem to maintain an upper bound on the utility of an arm based on the observed outcomes of when other arms are played. The next lemma characterizes these upper bounds on the utility called the pseudo empirical estimates.

Lemma 3. *Given any bandit instance I, s_{im}, the pseudo empirical estimate of arm i with respect to arm m is upper bounded such that*

$$s_{im} \leq (\sum_{l=1}^{m}\mu_l+(i-m)\mu_m+c_m-c_i)/i.$$

Proof. We assume that the expected utilities of the tasks is a decreasing function and the cost incurred an increasing function. The expected reward or utility by pulling configuration i can be written as

$$\hat{f}_i \leq (\sum_{l=1}^{i}\mu_l-c_i)/i$$

Let us take the case where $m<i.$ Using the decreasing property of expected utilities, we have,

$$i\,\hat{f}_i \leq \sum_{l=1}^{m}\mu_l+(i-m)\mu_m-c_i = m\,\hat{f}_m+(i-m)\mu_m-c_i+c_m$$

Let us now take the case where $m>i.$

$$i\,\hat{f}_i = \sum_{l=1}^{m}\mu_l-\sum_{l=i+1}^{m}\mu_l-c_i = m\,\hat{f}_m+c_m-\sum_{l=i+1}^{m}\mu_l-c_i$$

Using the decreasing property of expected utilities, we get

$$\leq m\,\hat{f}_m+c_m-c_i-(m-i)\mu_m$$

The formal definition of U-CUCB in given in Algorithm 2. In Figure 1, we compare the performance of U-UCB with U-CUCB over 100 iterations where in each iteration the rewards and costs are sampled randomly and sorted so as to satisfy the definitions 2 and 3. One example instance for of parameters for our simulation setup is; number of arms $M=10$, reward vector $\mu=$ [0.99816519, 0.96669968, 0.95893375, 0.95473951, 0.80981987, 0.61577662, 0.56091923, 0.44795704, 0.21987076, 0.18524995], cost vector $c=$ [0.0845928, 0.13959031, 0.17494821, 0.29012894, 0.33420872, 0.47049682, 0.48849441, 0.71938964, 0.87916942, 0.98123324]. The value of time horizon T are given in Figure 1. We observe that the performance of the two algorithms is very close. Given that U-UCB is computationally less expensive than U-CUCB, we thus conclude that comparable performance can be obtained with a simpler policy that does not exploit correlated across arms.

Algorithm 2: UTILITY-CORRELATED UCB (U-CUCB)

```
Input: $c_m$ for $1 \le m \le M$
Initialise: Set $\hat{\mu}_m = 0, n_m = 0$ and $\eta_m = 0$
  $\forall m \in [M],\ s_{z,m} = \phi_{z,m} = 0,\ \forall m, z \in [M]$
  $r = 1, t = 1$
While $r > 0$ do
    $k_{\max} = \arg\max_{m \in [M]} \eta_m$
    $\mathrm{A} = [M] \setminus k_{\max}$
    for $z \in \mathrm{A}$ do
        $\mathrm{A} = \mathrm{A} \setminus z$ if $\frac{\sum_{j=1}^{k_{\max}} \hat{\mu}_j - c_{k_{\max}}}{k_{\max}} > \phi_{z,k_{\max}}$
    for $m \in [M]$ do
        $\hat{\gamma}_m = \frac{\sum_{l=1}^{m} \hat{\mu}_l - c_m}{m} + \sqrt{\frac{2\log t}{m \times n_m}}$
     Choose arm $j$, where
        $j = \begin{cases} 1 & \text{if r = 1} \\ M & \text{if r = 2} \\ \arg\max_{m \in [\mathrm{A}]} \hat{\gamma}_m & \text{otherwise}. \end{cases}$
        $n_j = n_j + 1$
        for $k \in [j]$ do
            Receive reward $X_k \sim \mathrm{Ber}(\mu_k)$
            $\hat{\mu}_k = (\hat{\mu}_k \cdot n_k + X_k)/(n_k + 1)$
            $n_k = n_k + 1$
        $r = r + 1,\ t = t + j$
        for $z \in [M] \setminus j$ do
            $s_{z,j} = (\sum_{l=1}^{m} \hat{\mu}_l + (z - j)\hat{\mu}_l + c(j) - c(z))/z$
            $\phi_{z,j} = \phi_{z,j} + \frac{s_{z,j}}{\eta(j)}$
```

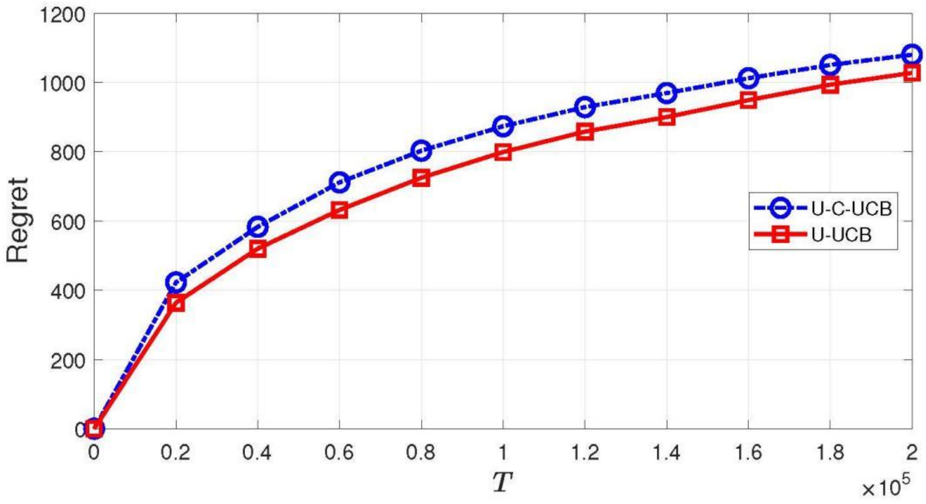

Figure 1. Regret as a function of time.

Acknowledgments

This work is supported by a SERB grant on Leveraging Edge Resources for Service Hosting and a SERB grant on Versatile Scheduling Policies with Provable Guarantees for Latency Sensitive Systems.

Conflict of Interest

There is no conflict of interest for this study.

References

[1] Auer, P.; Cesa-Bianchi, N.A.; Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* **2002**, *47*, 235–256, https://doi.org/10.1023/a:1013689704352.

[2] Tang, H.; Ciblat, P.; Wang, J.; Wigger, M.; Yates, R. Age of information aware cache updating with file- and age-dependent update durations. In Proceedings of 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), Volos, Greece, 15–19 June 2020. pp. 1–6.

[3] Ahani, G.; Yuan, D.; Sun, S. Optimal Scheduling of Age-centric Caching: Tractability and Computation. *IEEE Trans. Mob. Comput.* **2020**, *21*, 2939–2954. https://doi.org/10.1109/tmc.2020.3045104.

[4] Ma, M.; Wong, V.W. Age of Information Driven Cache Content Update Scheduling for Dynamic Contents in Heterogeneous Networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 8427–8441, https://doi.org/10.1109/twc.2020.3022895.

[5] Wu, X.; Li, X.; Li, J.; Ching, P.C.; Poor, H.V. Deep Reinforcement Learning for loT Networks: Age of Information and Energy Cost Tradeoff. In Proceedings of GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020, https://doi.org/10.1109/globecom42002.2020.9322415.

[6] Bhandari, K.; Fatale, S.; Narula, U.; Moharir, S.; Hanawal, M.K. Age-of-information bandits. In Proceedings of 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), Volos, Greece, 15–19 June 2020. pp. 1–8.

[7] Atay, E.U.; Kadota, I.; Modiano, E. Aging bandits: Regret analysis and order-optimal learning algorithm for wireless networks with stochastic arrivals. arXiv preprint arXiv:2012.08682, 2020.

[8] Juneja, I.; Fatale, S.; Moharir, S. Correlated age-of-information bandits. In Proceedings of 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March 2021–1 April 2021, https://doi.org/10.1109/WCNC49053.2021.9417327.

[9] Prasad, A.; Jain, V.; Moharir, S. Decentralized age-of-information bandits. In Proceedings of 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March 2021–1 April 2021, https://doi.org/10.1109/WCNC49053.2021.9417301.

[10] Li, B. Efficient Learning-based Scheduling for Information Freshness in Wireless Networks. In Proceedings of IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021, https://doi.org/10.1109/infocom42981.2021.9488709.

[11] Gupta, S.; Chaudhari, S.; Mukherjee, S.; Joshi, G.; Yağan, O. A unified approach to translate classical bandit algorithms to the structured bandit setting. In Proceedings of ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021, https://doi.org/10.1109/ICASSP39728.2021.9413628.

[12] Mersereau, A.J.; Rusmevichientong, P.; Tsitsiklis, J.N. A structured multiarmed bandit problem and the greedy policy. *IEEE Trans. Automat. Contr.* **2009**, *54*, 2787–2802. https://doi.org/ 10.1109/TAC.2009.2031725

[13] Hong, J.; Kveton, B.; Zaheer, M.; Chow, Y.; Ahmed, A.; Boutilier, C. Latent bandits revisited. *Adv. Neural. Inf. Process. Syst.* **2020**, *33*, 13423–13433.

[14] Gupta, S.; Joshi, G.; Yağan, O. Best-arm identification in correlated multi-armed bandits. *IEEE J. Sel. Areas Inf. Theory* **2021**, *2*, 549–563, https://doi.org/10.1109/JSAIT.2021.3082028

[15] Soare, M.; Lazaric, A.; Munos, R. Best-arm identification in linear bandits. *Adv. Neural. Inf. Process. Syst.* **2014**, *27*, 828–836.

[16] Tao, C.; Blanco, S.; Zhou, Y. Best arm identification in linear bandits with linear dimension dependency. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018. pp. 4877–4886.

[17] Jamieson, K.; Nowak, R. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In Proceedings of 2014 48th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 19–21 March 2014, https://doi.org/10.1109/ciss.2014.6814096.

[18] Lattimore, T.; Szepesvári, C. *Bandit algorithms*. Cambridge University Press: Cambridge, England, 2020.

[19] Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86, https://doi.org/10.1214/aoms/1177729694.

[20] Lai, T.; Robbins, H. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* **1985**, *6*, 4–22, https://doi.org/10.1016/0196-8858(85)90002-8.

[21] Gupta, S.; Joshi, G.; Yağan, O. Correlated multi-armed bandits with a latent random source. In Proceedings of ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Barcelona, Spain, 4–8 May 2020, https://doi.org/10.1109/ICASSP40776.2020.9054429.