Combinatorial Bandits for Incentivizing Agents with Dynamic Preferences

Tanner Fiez[*], **Shreyas Sekar**[*], **Liyuan Zheng,** and **Lillian J. Ratliff**
Electrical Engineering Department, University of Washington

Abstract

The design of personalized incentives or recommendations to improve user engagement is gaining prominence as digital platform providers continually emerge. We propose a multi-armed bandit framework for matching incentives to users, whose preferences are unknown *a priori* and evolving dynamically in time, in a resource constrained environment. We design an algorithm that combines ideas from three distinct domains: (i) a greedy matching paradigm, (ii) the upper confidence bound algorithm (UCB) for bandits, and (iii) mixing times from the theory of Markov chains. For this algorithm, we provide theoretical bounds on the regret and demonstrate its performance via both synthetic and realistic (matching supply and demand in a bike-sharing platform) examples.

1 INTRODUCTION

The theory of *multi-armed bandits* plays a key role in enabling personalization in the digital economy (Scott, 2015). Algorithms from this domain have successfully been deployed in a diverse array of applications including online advertising (Lu et al., 2010; Mehta and Mirrokni, 2011), crowdsourcing (Tran-Thanh et al., 2014), content recommendation (Li et al., 2010), and selecting user-specific incentives (Ghosh and Hummel, 2013; Jain et al., 2014) (e.g., a retailer offering discounts). On the theoretical side, this has been complemented by a litany of *near-optimal regret bounds* for multi-armed bandit settings with rich combinatorial structures and complex agent behavior models (Chen et al., 2016; Gai et al., 2011; Kveton et al., 2015; Sani et al., 2012). At a high level, the broad appeal of bandit approaches for allocating resources to human agents stems from its focus on balancing *exploration* with *exploitation*, thereby allowing a decision-maker to efficiently identify users' preferences without sacrificing short-term rewards.

Implicit in most of these works is the notion that in large-scale environments, a designer can simultaneously allocate resources to multiple users by running independent bandit instances. In reality, such independent decompositions do not make sense in applications where resources are subject to physical or monetary constraints. In simple terms, matching an agent to a resource immediately constrains the set of resources to which another agent can be matched. Such supply constraints may arise even when dealing with intangible products. For instance, social media platforms (e.g., Quora) seek to maximize user participation by offering incentives in the form of increased recognition—e.g., featured posts or badges (Immorlica et al., 2015). Of course, there are supply constraints on the number of posts or users that can be featured at a given time. As a consequence of these coupling constraints, much of the existing work on multi-armed bandits does not extend naturally to multi-agent economies.

Yet, another important aspect not addressed by the literature concerns human behavior. Specifically, users' preferences over the various resources may be dynamic—i.e. evolve in time as they are repeatedly exposed to the available options. The problem faced by a designer in such a dynamic environment is compounded by the lack of information regarding each user's current state or beliefs, as well as how these beliefs influence their preferences and their evolution in time.

Bearing in mind these limitations, we study a multi-armed bandit problem for matching multiple agents to a finite set of incentives[1]: each incentive belongs to a cate-

[*]Authors contributed equally.

[1]We use the term incentive broadly to refer to any resource or action available to the agent. That is, incentives are not limited to monetary or financial mechanisms.

gory and global capacity constraints control the number of incentives that can be chosen from each category. In our model, each agent has a *preference profile* or a *type* that determines its rewards for being matched to different incentives. The agent's type evolves according to a Markov decision process (MDP), and therefore, the rewards vary over time *in a correlated fashion*.

Our work is primarily motivated by the problem faced by a technological platform that seeks to not just maximize user engagement but also to encourage users to make changes in their *status quo* decision-making process by offering incentives. For concreteness, consider a bike-sharing service—an application we explore in our simulations—that seeks to identify optimal incentives for each user from a finite bundle of options—e.g., varying discount levels, free future rides, bulk ride offers, etc. Users' preferences over the incentives may evolve with time depending on their current type, which in turn depends on their previous experience with the incentives. In addition to their marketing benefits, such incentives can serve as a powerful instrument for *nudging* users to park their bikes at alternative locations—this can lead to spatially balanced supply and consequently, lower rejection rates (Singla et al., 2015).

1.1 CONTRIBUTIONS AND ORGANIZATION

Our objective is to design a multi-armed bandit algorithm that repeatedly matches agents to incentives in order to minimize the cumulative *regret* over a finite time horizon. Here, regret is defined as the difference in the reward obtained by a problem specific benchmark strategy and the proposed algorithm (see Definition 1). A preliminary impediment in achieving this goal is the fact that the capacitated matching problem studied in this work is NP-Hard even in the offline case. The major challenge therefore is whether *we can achieve sub-linear (in the length of the horizon) regret in the more general matching environment without any information on the users' underlying beliefs or how they evolve*?

Following preliminaries (Section 2), we introduce a simple greedy algorithm that provides a $1/3$–approximation to the optimal offline matching solution (Section 3). Leveraging this first contribution, the central result in this paper (Section 4) is a new multi-armed bandit algorithm—*MatchGreedy-EpochUCB* (MG-EUCB)—for capacitated matching problems with time-evolving rewards. Our algorithm obtains logarithmic (and hence sub-linear) regret even for this more general bandit problem. The proposed approach combines ideas from three distinct domains: (i) the $1/3$–rd approximate greedy matching algorithm, (ii) the traditional UCB algorithm (Auer et al., 2002), and (iii) mixing times from the theory of Markov chains.

We validate our theoretical results (Section 5) by performing simulations on both synthetic and realistic instances derived using data obtained from a Boston-based bike-sharing service *Hubway* (hub). We compare our algorithm to existing UCB-based approaches and show that the proposed method enjoys favorable convergence rates, computational efficiency on large data sets, and does not get stuck at sub-optimal matching solutions.

1.2 BACKGROUND AND RELATED WORK

Two distinct features separate our model from the majority of work on the multi-armed bandit problem: (i) our focus on a capacitated matching problem with finite supply (every user cannot be matched to their optimal incentive), and (ii) the rewards associated with each agent evolve in a correlated fashion but the designer is unaware of each agent's current state. Our work is closest to (Gai et al., 2011) which considers a matching problem with Markovian rewards. However, in their model the rewards associated with each edge evolve independently of the other edges; as we show via a simple example in Section 2.2, the correlated nature of rewards in our instance can lead to additional challenges and convergence to sub-optimal matchings if we employ a traditional approach as in (Gai et al., 2011).

Our work also bears conceptual similarities to the rich literature on combinatorial bandits (Badanidiyuru et al., 2013; Chen et al., 2016; Kveton et al., 2014, 2015; Wen et al., 2015). However, unlike our work, these papers consider a model where the distribution of the rewards is static in time. For this reason, efficient learning algorithms leveraging oracles to solve generic constrained combinatorial optimization problems developed for the combinatorial semi-bandit setting (Chen et al., 2016; Kveton et al., 2015) face similar limitations in our model as the approach of (Gai et al., 2011). Moreover, the rewards in our problem may not have a linear structure so the approach of (Wen et al., 2015) is not applicable.

The novelty in this work is not the combinatorial aspect but the interplay between combinatorial bandits and the edge rewards evolving according to an MDP. When an arm is selected by an oracle, the reward of every edge in the graph evolves—how it evolves depends on which arm is chosen. If the change occurs in a sub-optimal direction, this can affect future rewards. Indeed, the difficulties in our proofs do not stem from applying an oracle for combinatorial optimization, but from bounding the secondary regret that arises when rewards evolve in a sub-optimal way.

Finally, there is a somewhat parallel body of work

on single-agent reinforcement learning techniques (Azar et al., 2013; Jaksch et al., 2010; Mazumdar et al., 2017; Ratliff et al., 2018) and expert selection where the rewards on the arms evolve in a correlated fashion as in our work. In addition to our focus on multi-agent matchings, we remark that many of these works assume that the designer is aware (at least partially) of the agent's exact state and thus, can eventually infer the nature of the evolution. Consequently, a major contribution of this work is the extension of UCB-based approaches to solve MDPs with a *fully unobserved state* and rewards associated with each edge that evolve in a correlated fashion.

2 PRELIMINARIES

A designer faces the problem of matching m agents to incentives (more generally jobs, goods, content, etc.) without violating certain capacity constraints. We model this setting by means of a bipartite graph $(\mathcal{A}, \mathcal{I}, \mathcal{P})$ where $\mathcal{A}$ is the set of agents, $\mathcal{I}$ is the set of incentives to which the agents can be matched, and $\mathcal{P} = \mathcal{A} \times \mathcal{I}$ is the set of all pairings between agents and incentives. We sometimes refer to $\mathcal{P}$ as the set of arms. In this regard, a matching is a set $M \subseteq \mathcal{P}$ such that every agent $a \in \mathcal{A}$ and incentive $i \in \mathcal{I}$ is present in at most one edge belonging to M.

Each agent $a \in \mathcal{A}$ is associated with a type or state $\theta_a \in \Theta_a$, which influences the reward received by this agent when matched with some incentive $i \in \mathcal{I}$. When agent a is matched to incentive i, its type evolves according to a Markov process with transition probability kernel $P_{a,i} : \Theta_a \times \Theta_a \to [0, 1]$. Each pairing or edge of the bipartite graph is associated with some reward that depends on the agent–incentive pair, (a, i), as well as the type θ_a.

The designer's policy (algorithm) is to compute a matching repeatedly over a finite time horizon in order to maximize the expected aggregate reward. In this work, we restrict our attention to a specific type of multi-armed bandit algorithm that we refer to as an *epoch mixing policy*. Formally, the execution of such a policy α is divided into a finite number of time indices $[n] = \{1, 2, \ldots, n\}$, where n is the length of the time horizon. In each time index $k \in [n]$, the policy selects a matching $\alpha(k)$ and repeatedly 'plays' this matching for $\tau_k > 0$ iterations within this time index. We refer to the set of iterations within a time index collectively as an *epoch*. That is, within the k–th epoch, for each edge $(a, i) \in \alpha(k)$, agent a is matched to incentive i and the agent's type is allowed to evolve for τ_k iterations. In short, an epoch mixing policy proceeds in two time scales—each selection of a matching corresponds to an epoch comprising of τ_k iterations for $k \in [n]$, and there are a total of n epochs. It is worth noting that an epoch-based policy was used in the UCB2 algorithm (Auer et al., 2002), albeit with stationary rewards.

Agents' types evolve based on the incentives to which they are matched. Suppose that $\beta_a^{(k)}$ denotes the type distribution on Θ_a at epoch k and $i \in \mathcal{I}$ is the incentive to which agent a is matched by α (i.e., $(a, i) \in \alpha(k)$). Then, $\beta_a^{(k+1)}(\theta_a) = \sum_{\theta' \in \Theta_a} P_{a,i}^{\tau_k}(\theta', \theta_a) \beta_a^{(k)}(\theta')$.

For epoch k, the rewards are averaged over the τ_k iterations in that epoch. Let $r_{a,i}^{\theta}$ denote the reward received by agent a when it is matched to incentive i given type $\theta \in \Theta_a$. We assume that $r_{a,i}^{\theta} \in [0, 1]$ and is drawn from a distribution $\mathcal{T}_r(a, i, \theta)$. The reward distributions for different edges and states in Θ_a are assumed to be independent of each other. Suppose that an algorithm α selects the edge (a, i) for τ iterations within an epoch. The observed reward at the end of this epoch is taken to be the time-averaged reward over the epoch. Specifically, suppose that the k–th epoch proceeds for τ_k iterations beginning with time t_k—i.e. one plus the total iterations completed before this—and ending at time $t_{k+1} - 1 = t_k + \tau_k - 1$, and that $\theta_a(t)$ denotes agent a's state at time $t \in [t_k, t_{k+1} - 1]$. Then, the time-averaged reward in the epoch is given by $\boldsymbol{r}_{a,i}^{\theta_a(t_k)} = \frac{1}{\tau_k} \sum_{t=t_k}^{t_{k+1}-1} r_{a,i}^{\theta_a(t)}$. We use the state as a superscript to denote dependence of the reward on the agent's type at the beginning of the epoch. Finally, the total (time-averaged) reward due to a matching $\alpha(k)$ at the end of an epoch can be written as $\sum_{(a,i) \in \alpha(k)} \boldsymbol{r}_{a,i}^{\theta_a(t_k)}$.

We assume that the Markov chain corresponding to each edge $(a, i) \in \mathcal{P}$ is *aperiodic* and *irreducible* (Levin et al., 2009). We denote the stationary or steady-state distribution for this edge as $\pi_{a,i} : \Theta_a \to [0, 1]$. Hence, we define the expected reward for edge (a, i), given its stationary distribution, to be $\mu_{a,i} = \mathbb{E}\left[\sum_{\theta \in \Theta_a} r_{a,i}^{\theta} \pi_{a,i}(\theta)\right]$ where the expectation is with respect to the distribution on the reward given θ.

2.1 CAPACITATED MATCHING

Given $\mathcal{P} = \mathcal{A} \times \mathcal{I}$, the designer's goal at the beginning of each epoch is to select a matching $M \subseteq \mathcal{P}$—i.e. a collection of edges—that satisfies some cardinality constraints. We partition the edges in $\mathcal{P}$ into a mutually exclusive set of classes allowing for edges possessing identical characteristics to be grouped together. In the bike-sharing example, the various classes could denote types of incentives—e.g., edges that match agents to discounts, free-rides, etc. Suppose that $\mathcal{C} = \{\xi_1, \xi_2, \ldots, \xi_q\}$ denotes a partitioning of the edge set such that (i) $\xi_j \subseteq \mathcal{P}$ for all $1 \leq j \leq q$, (ii) $\bigcup_{j=1}^{q} \xi_j = \mathcal{P}$, and (iii) $\xi_j \cap \xi_{j'} = \emptyset$ for all $j \neq j'$. We refer to each ξ_j as a class and for any given edge $(a, i) \in \mathcal{P}$, use $c(a, i)$ to denote the class that this edge belongs to, i.e., $(a, i) \in c(a, i)$ and $c(a, i) \in \mathcal{C}$.

Given a capacity vector $\boldsymbol{b} = (b_{\xi_1}, \ldots, b_{\xi_q})$ indexed on the set of classes, we say that a matching $M \subseteq \mathcal{P}$ is a feasible solution to the capacitated matching problem if:

a) for every $a \in \mathcal{A}$ (resp., $i \in \mathcal{I}$), the matching must contain at most one edge containing this agent (resp., incentive)
b) and, the total number of edges from each class ξ_j contained in the matching cannot be larger than b_{ξ_j}.

In summary, the *capacitated matching problem* can be formulated as the following integer program:

$$\begin{aligned} \max \quad & \textstyle\sum_{(a,i)\in\mathcal{P}} w(a,i)x(a,i) \\ \text{s.t.} \quad & \textstyle\sum_{i\in\mathcal{I}} x(a,i) \le 1 \quad \forall a \in \mathcal{A} \\ & \textstyle\sum_{a\in\mathcal{A}} x(a,i) \le 1 \quad \forall i \in \mathcal{I} \\ & \textstyle\sum_{(a,i)\in\xi_j} x(a,i) \le b_{\xi_j}, \quad \forall \xi_j \in \mathcal{C} \\ & x(a,i) \in \{0,1\}, \quad \forall (a,i) \in \mathcal{P} \end{aligned} \tag{P1}$$

We use the notation $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$ for a *capacitated matching problem instance*. In (P1), $w(a,i)$ refers to the weight or the reward to be obtained from the given edge. The term $x(a,i)$ is an indicator on whether the edge (a,i) is included in the solution to (P1). Clearly, the goal is to select a maximum weight matching subject to the constraints. In our online bandit problem, the designer's actual goal in a fixed epoch k is to maximize the quantity $\sum_{(a,i)\in\mathcal{P}} \boldsymbol{r}_{a,i}^{\theta_a(t_k)} x(a,i)$, i.e., $w(a,i) = \boldsymbol{r}_{a,i}^{\theta_a(t_k)}$. However, since the reward distributions and the current user type are not known beforehand, our MG-EUCB algorithm (detailed in Section 4.2) approximates this objective by setting the weights to be the average observed reward from the edges in combination with the corresponding confidence bounds.

2.2 TECHNICAL CHALLENGES

There are two key obstacles involved in extending traditional bandit approaches to our combinatorial setting with evolving rewards, namely, *cascading sub-optimality* and *correlated convergence*. The first phenomenon occurs when an agent a is matched to a sub-optimal arm i (incentive) because its optimal arm i^* has already been assigned to another agent. Such sub-optimal pairings have the potential to cascade, e.g., when another agent a_1 who is matched to i in the optimal solution can no longer receive this incentive and so on. Therefore, unlike the classical bandit analysis, the selection of sub-optimal arms cannot be directly mapped to the empirical rewards.

Correlated Convergence. As mentioned previously, in our model, the rewards depend on the type or state of an agent, and hence, the reward distribution on any given edge (a,i) can vary even when the algorithm does not select this edge. As a result, a naïve application of a bandit algorithm can severely under-estimate the expected reward on each edge and eventually converge to a sub-optimal matching. A concrete example of the poor convergence effect is provided in Example 1. In Section 4.2, we describe how our central bandit algorithm limits the damage due to cascading while simultaneously avoiding the correlated convergence problem.

Example 1 (Failure of Classical UCB)**.** *Consider a problem instance with two agents $\mathcal{A} = \{a_1, a_2\}$, two incentives $\mathcal{I} = \{i_1, i_2\}$ and identical state space i.e., $\Theta_{a_1} = \Theta_{a_2} = \{\theta_1, \theta_2\}$. The transition matrices and deterministic rewards for the agents for being matched to each incentive are depicted pictorially below: we assume that $\epsilon > 0$ is a sufficiently small constant.*

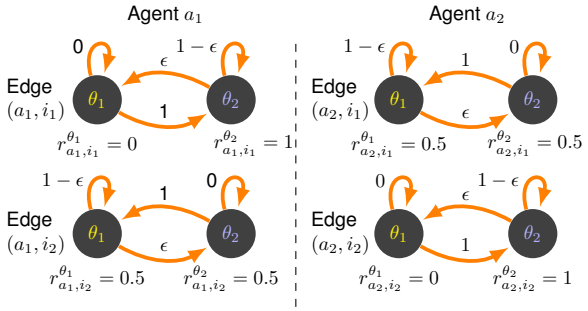


Figure 1: (a) State transition diagram and reward for each edge: note that the state is associated with the agent and not the edge.

Clearly, the optimal strategy is to repeatedly chose the matching $\{(a_1, i_1), (a_2, i_2)\}$ achieving a reward of (almost) two in each epoch. An implementation of traditional UCB for the matching problem—e.g., the approach in (Chen et al., 2016; Gai et al., 2011; Kveton et al., 2015)—selects a matching based on the empirical rewards and confidence bounds for a total of $\sum_{k=1}^{n} \tau_k$ iterations, which are then divided into epochs for convenience. This approach converges to the sub-optimal matching of $M = \{(a_1, i_2), (a_2, i_1)\}$. Indeed, every time the algorithm selects this matching, both the agents' states are reset to θ_1 and when the algorithm explores the optimum matching, the reward consistently happens to be zero since the agents are in state θ_1. Hence, the rewards for the (edges in the) optimum matching are grossly underestimated.

3 GREEDY OFFLINE MATCHING

In this section, we consider the capacitated matching problem in the offline case, where the edge weights are provided as input. The techniques developed in this section serve as a base in order to solve the more general online problem in the next section. More specifically, we assume that we are given an arbitrary instance of the capacitated matching problem $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$.

Algorithm 1 Capacitated-Greedy Matching Algorithm

function MG($(w(a,i))_{(a,i)\in\mathcal{P}}$, $\boldsymbol{b}$)
  $G^* \leftarrow \emptyset$, $E' \leftarrow \mathcal{P}$
  while $E' \neq \emptyset$:
    Select $(a,i) = \arg\max_{(a',i')\in E'} w(a',i')$
    if $|G^* \cap c(a,i)| < b_{c(a,i)}$ **then**
      $G^* \leftarrow G^* \cup \{(a,i)\}$
      $E' \leftarrow E' \setminus \{(a',i')\}\ \forall (a',i') : a' = a$ or $i' = i$
    else
      $E' \leftarrow E' \setminus \{(a,i)\}$
  return G^*
end function

Given this instance, the designer's objective is to solve (P1). Surprisingly, this problem turns out to be NP-Hard and thus cannot be optimally solved in polynomial time (Garey and Johnson, 1979)—this marks a stark contrast with the classic maximum weighted matching problem, which can be solved efficiently using the Hungarian method (Kuhn, 1955).

In view of these computational difficulties, we develop a simple greedy approach for the capacitated matching problem and formally prove that it results in a one-third approximation to the optimum solution. The greedy method studied in this work comes with a multitude of desirable properties that render it suitable for matching problems arising in large-scale economies. Firstly, the greedy algorithm has a running time of $O(m^2 \log m)$, where m is the number of agents—this near-linear execution time in the number of edges makes it ideal for platforms comprising of a large number of agents. Secondly, since the output of the greedy algorithm depends only on the ordering of the edge weights and is not sensitive to their exact numerical value, learning approaches tend to converge faster to the 'optimum solution'. This property is validated by our simulations (see Figure 2c). Finally, the performance of the greedy algorithm in practice (e.g., see Figure 2b) appears to be much closer to the optimum solution than the 1/3 approximation guaranteed by Theorem 1 below.

3.1 ANALYSIS OF GREEDY ALGORITHM

The greedy matching is outlined in Algorithm 1. Given an instance $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$, Algorithm 1 'greedily' selects the highest weight feasible edge in each iteration—this step is repeated until all available edges that are feasible are added to G^*. Our main result in this section is that for any given instance of the capacitated matching problem, the matching G^* returned by Algorithm 1 has a total weight that is at least 1/3–rd that of the maximum weight matching.

Theorem 1. *For any given capacitated matching problem instance $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$, let G^* denote the output of Algorithm 1 and M^* be any other feasible solution to the optimization problem in* (P1) *including the optimum matching. Then, $\sum_{(a,i)\in M^*} w(a,i) \leq 3\sum_{(a,i)\in G^*} w(a,i)$.*

The proof is based on a *charging argument* that takes into account the capacity constraints and can be found in Section B.1 of the supplementary material. At a high level, we take each edge belonging to the benchmark M^* and identify a corresponding edge in G^* whose weight is larger than that of the benchmark edge. This allows us to charge the weight of the original edge to an edge in G^*. During the charging process, we ensure that no more than three edges in M^* are charged to each edge in G^*. This gives us an approximation factor of three.

3.2 PROPERTIES OF GREEDY MATCHINGS

We conclude this section by providing a hierarchical decomposition of the edges in $\mathcal{P}$ for a fixed instance $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$. In Section 4.1, we will use this property to reconcile the offline version of the problem with the online bandit case. Let $G^* = \{g_1^*, g_2^*, \ldots, g_m^*\}$ denote the matching computed by Algorithm 1 for the given instance such that $w(g_1^*) \geq w(g_2^*) \geq \ldots \geq w(g_m^*)$ without loss of generality[2]. Next, let $G_j^* = \{g_1^*, g_2^*, \ldots, g_j^*\}$ for all $1 \leq j \leq m$—i.e. the j highest-weight edges in the greedy matching.

For each $1 \leq j \leq m$, we define the infeasibility set $H_j^{G^*}$ as the set of edges in $\mathcal{P}$ that when added to G_j^* violates the feasibility constraints of (P1). Finally, we use $L_j^{G^*}$ to denote the marginal infeasibility sets—i.e. $L_1^{G^*} = H_1^{G^*}$ and

$$L_j^{G^*} = H_j^{G^*} \setminus H_{j-1}^{G^*},\ \forall\, 2 \leq j \leq m. \tag{1}$$

We note that the marginal infeasibility sets denote a mutually exclusive partition of the edge set minus the greedy matching—i.e., $\bigcup_{1\leq j\leq m} L_j^{G^*} = \mathcal{P} \setminus G^*$. Moreover, since the greedy matching selects its edges in the decreasing order of weight, for any $g_j^* \in G^*$, and every $(a,i) \in L_j^{G^*}$, we have that $w(g_j^*) \geq w(a,i)$.

Armed with our decomposition of the edges in $\mathcal{P} \setminus G^*$, we now present a crucial structural lemma. The following lemma identifies sufficient conditions on the *local ordering* of the edge weights for two different instances under which the outputs of the greedy matching for the instances are non-identical.

Lemma 1. *Given instances $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$ and $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (\tilde{w}(a,i))_{(a,i)\in\mathcal{P}}\}$ of the capacitated matching problem, let $G^* = \{g_1^*, g_2^*, \ldots, g_m^*\}$ and $\tilde{G}$ denote*

[2] If $g = (a,i)$, we abuse notation and let $w(g) = w(a,i)$.

the output of Algorithm 1 for these instances, respectively. Let E_1, E_2 be conditions described as follows:

$$E_1 = \{\exists j < j' \,|(\tilde{w}(g_j^*) < \tilde{w}(g_{j'}^*)) \wedge (g_{j'}^* \in \tilde{G})\}$$
$$E_2 = \{\exists g_j^* \in G^*, (a,i) \in L_j^{G^*} \mid (\tilde{w}(g_j^*) < \tilde{w}(a,i)) \wedge ((a,i) \in \tilde{G})\}.$$

If $G^* \neq \tilde{G}$, then at least one of E_1 or E_2 must be true.

Lemma 1 is fundamental in the analysis of our MG-EUCB algorithm because it provides a method to map the selection of each sub-optimal edge to a familiar condition comparing empirical rewards to stationary rewards.

4 ONLINE MATCHING—BANDIT ALGORITHM

In this section, we propose a multi-armed bandit algorithm for the capacitated matching problem and analyze its regret. For concreteness, we first highlight the information and action sets available to the designer in the online problem. The designer is presented with a *partial instance* of the matching problem without the weights, i.e., $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}\}$ along with a fixed time horizon of n epochs but has the ability to set the parameters $(\tau_1, \tau_2, \ldots, \tau_n)$, where τ_k is the number of iterations under epoch k. The designer's goal is to design a policy α that selects a matching $\alpha(k)$ in the k–th epoch that is a feasible solution for (P1). At the end of the k–th epoch, the designer observes the average reward $\boldsymbol{r}_{a,i}^{\theta_a(k)}$ for each $(a,i) \in \alpha(k)$ but *not the agent's type*. We abuse notation and take $\theta_a(k)$ to be the agent's state at the beginning of epoch k. The designer's objective is to minimize the regret over the finite horizon.

The expected regret of a policy α is the difference in the expected aggregate reward of a benchmark matching and that of the matching returned by the policy, summed over n epochs. Owing to its favorable properties (see Section 3), we use the greedy matching on the stationary state rewards as our benchmark. Measuring the regret with respect to the unknown stationary-distribution is standard with MDPs (e.g., see (Gai et al., 2011; Tekin and Liu, 2010, 2012)). Formally, let G^* denote the output of Algorithm 1 on the instance $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (\mu_{a,i})_{(a,i)\in\mathcal{P}}\}$—i.e., with the weights $w(a,i)$ equal the stationary state rewards $\mu_{a,i}$.

Definition 1. *The expected regret of a policy α with respect to the greedy matching G^* is given by*

$$R^\alpha(n) = n \sum_{(a,i)\in G^*} \mu_{a,i} - \sum_{k=1}^{n} \sum_{(a,i)\in\alpha(k)} \mathbb{E}[\boldsymbol{r}_{a,i}^{\theta_a(k)}],$$

where the expectation is with respect to the reward and the state of the agents during each epoch.

4.1 REGRET DECOMPOSITION

As is usual in this type of analysis, we start by decomposing the regret in terms of the number of selections of each sub-optimal arm (edge). We state some assumptions and define notation before proving our generic regret decomposition theorem. A complete list of the notation used can be found in Section A of the supplementary material.

1. For analytic convenience, we assume that the number of agents and incentives is balanced and therefore, $|\mathcal{A}| = |\mathcal{I}| = m$. WLOG, every agent is matched to some incentive in G^*; if this is not the case, we can add *dummy incentives* with zero reward.
2. Suppose that $G^* = \{g_1^*, g_2^*, \ldots, g_m^*\}$ such that $\mu_{g_1^*} \geq \ldots \geq \mu_{g_m^*}$ and let $i^*(a)$ denote the incentive that a is matched to in G^*. Let $L_1^*, \ldots L_m^*$ be the marginal infeasibility sets as defined in (1).
3. Suppose that $\tau_0 \geq 1$ and $\tau_k = \tau_0 + \zeta k$ for some non-negative integer ζ.

Let $\mathbb{1}\{\cdot\}$ be the indicator function—e.g., $\mathbb{1}\{(a,i) \in \alpha(k)\}$ is one when the edge (a,i) belongs to the matching $\alpha(k)$, and zero otherwise. Define $T_{a,i}^\alpha(n) = \sum_{k=1}^n \mathbb{1}\{(a,i) \in \alpha(k)\}$ to be the random variable that denotes the number of epochs in which an edge is selected under an algorithm α. By relating $\mathbb{E}[T_{a,i}^\alpha(n)]$ to the regret $R^\alpha(n)$, we are able to provide bounds on the performance of α.

By adding and subtracting $\sum_{(a,i)\in\mathcal{P}} \mathbb{E}[T_{a,i}^\alpha(n)]\mu_{a,i}$ from the equation in Definition 1, we get that

$$R^\alpha(n) = \sum_{(a,i)\in\mathcal{P}} \mathbb{E}[T_{a,i}^\alpha(n)](\mu_{a,i^*(a)} - \mu_{a,i}) + \sum_{k=1}^n \sum_{(a,i)\in\mathcal{P}} \mathbb{E}[\mathbb{1}\{(a,i) \in \alpha(k)\}\big(\mu_{a,i} - \boldsymbol{r}_{a,i}^{\theta_a(k)}\big)].$$

To further simplify the regret, we separate the edges in $\mathcal{P}$ by introducing the notion of a sub-optimal edge. Formally, for any given $a \in \mathcal{A}$, define $S_a := \{(a,i) \mid \mu_{a,i^*(a)} \geq \mu_{a,i}\ \forall i \in \mathcal{I}\}$ and $\mathcal{S} := \bigcup_{a\in\mathcal{A}} S_a$. Then, the regret bound in the above equation can be simplified by ignoring the contribution of the terms in $\mathcal{P} \setminus \mathcal{S}$. That is, since $\mu_{a,i^*(a)} < \mu_{a,i}$ for all $(a,i) \in \mathcal{P} \setminus \mathcal{S}$,

$$R^\alpha(n) \leq \sum_{(a,i)\in\mathcal{S}} \mathbb{E}[T_{a,i}^\alpha(n)](\mu_{a,i^*(a)} - \mu_{a,i}) + \sum_{k=1}^n \sum_{(a,i)\in\mathcal{P}} \mathbb{E}[\mathbb{1}\{(a,i) \in \alpha(k)\}\big(\mu_{a,i} - \boldsymbol{r}_{a,i}^{\theta_a(k)}\big)]. \quad (2)$$

Recall from the definition of the marginal infeasibility sets in (1) that for any given $(a,i) \in \mathcal{P} \setminus G^*$, there exists a unique edge $g_j^* \in G^*$ such that $(a,i) \in L_j^*$. Define $L^{-1}(a,i) := g_j^* \in G^*$ such that $(a,i) \in L_j^*$. Now, we can define the reward gap for any given edge as follows:

$$\Delta_{a,i} = \begin{cases} \mu_{a,i^*(a)} - \mu_{a,i}, & \text{if } (a,i) \in \mathcal{S} \\ \mu_{L^{-1}(a,i)} - \mu_{a,i}, & \text{if } (a,i) \in (\mathcal{P} \setminus G^*) \setminus \mathcal{S} \\ \mu_{g_{j-1}^*} - \mu_{g_j^*}, & \text{if } (a,i) = g_j^* \text{ for } j \geq 2 \end{cases}$$

This leads us to our main regret decomposition result which leverages mixing times for Markov chains (Fill, 1991) along with Equation (2) in deriving regret bounds. For an aperiodic, irreducible Markov chain $P_{a,i}$, using the notion that it convergences to its stationary state under repeated plays of a fixed action, we can prove that for every arm (a,i), there exists a constant $C_{a,i} > 0$ such that $\left|\mathbb{E}\left[\mu_{a,i} - \boldsymbol{r}_{a,i}^{\theta_a(k)}\right]\right| \leq C_{a,i}/\tau_k$—in fact, this result holds for all type distributions $\beta_a^{(k)}$ of the agent.

Proposition 1. *Suppose for each $(a,i) \in \mathcal{P}$, $P_{a,i}$ is an aperiodic, irreducible Markov chain with corresponding constant $C_{a,i}$. Then, for a given algorithm α where $\tau_k = \tau_0 + \zeta k$ for some fixed $\zeta > 0$, we have that*

$$R^{\alpha}(n) \leq \sum_{(a,i)\in\mathcal{S}} \mathbb{E}_{\alpha}\left[T_{a,i}^{\alpha}(n)\right](\Delta_{a,i} + \tfrac{C_{a,i}}{\tau_0}) + m\tfrac{C_*}{\zeta}\left(1 + \log\left(\zeta(n-1)/\tau_0 + 1\right)\right).$$

The proof of this proposition is in Section B.2 of the supplementary material.

4.2 MG-EUCB ALGORITHM AND ANALYSIS

In the initialization phase, the algorithm computes and plays a sequence of matchings $M_1, M_2, \ldots, M_p$ for a total of p epochs. The initial matchings ensure that every edge in $\mathcal{P}$ is selected at least once—the computation of these initial matchings relies on a *greedy covering* algorithm that is described in Section C.1 of the supplementary material. Following this, our algorithm maintains the cumulative empirical reward $\bar{r}_{a,i}$ for every $(a,i) \in \mathcal{P}$. At the beginning of (say) epoch k, the algorithm computes a greedy matching for the instance $\{\mathcal{P}, \mathcal{C}, \boldsymbol{b}, (w(a,i))_{(a,i)\in\mathcal{P}}\}$ where $w(a,i) = \bar{r}_{a,i}/T_{a,i} + c_{a,i}$, i.e., the average empirical reward for the edge added to a suitably chosen confidence window. The INCENT$(\cdot)$ function (Algorithm 2, described in the supplementary material since it is a trivial function) plays each edge in the greedy matching for τ_k iterations, where τ_k increases linearly with k. This process is repeated for $n-p$ epochs. Prior to theoretically analyzing MG-EUCB, we return to Example 1 in order to provide intuition for how the algorithm overcomes correlated convergence of rewards.

Revisiting Example 1: Why does MG-EUCB work? In Example 1, the algorithm initially estimates the empirical reward of (a_1, i_i) and (a_2, i_2) to be zero respectively. However, during the UCB exploration phase, the matching $M_1 = (a_1, i_1), (a_2, i_2)$ is played again for epoch length > 1 and the state of agent a_1 moves from θ_1 to θ_2 during the epoch. Therefore, the algorithm estimates the average reward of each edge within the epoch to be ≥ 0.5, and the empirical reward increases. This continues as the epoch length increases, so that eventually the empirical reward for (a_1, i_1) exceeds that of (a_1, i_2) and the algorithm correctly identifies the optimal matching as we move from exploration to exploitation.

Algorithm 2 MatchGreedy-EpochUCB

procedure MG-EUCB(ζ, τ_0, $\mathcal{P}$)
  $t_1 \leftarrow 0$, $\bar{r}_{a,i} \leftarrow 0$ & $T_{a,i} \leftarrow 1$ $\forall (a,i) \in \mathcal{P}$
  $M_1, M_2, \ldots, M_p \subset \mathcal{P}$ s.t. $(a,i) \in M_j \Leftrightarrow (a,i) \notin M_\ell$ $\forall \ell \neq j$ ▷ see Supplement C.1 for details
  INCENT$(\cdot)$ ▷ see Alg. 2 in Supplement C
  for $1 \leq n \leq m$ ▷ play each arm once
    $(\bar{r}_{a,i})_{(a,i)\in M_n} \leftarrow$ INCENT$(M_n, t_n, n, \tau_0, \zeta)$
    $t_{n+1} \leftarrow t_n + \tau_0 + \zeta n$
  end for
  while $n > m$
    $M_G = \text{MG}((\bar{r}_{a,i}/T_{a,i} + c_{a,i}^{T_{a,i}}(n))_{(a,i)\in\mathcal{P}})$
    $(r_{a,i}(t_n))_{(a,i)\in M_G} \leftarrow$ INCENT$(M_G, t_n, n, \tau_0, \zeta)$
    $\bar{r}_{a,i} \leftarrow \bar{r}_{a,i} + \frac{1}{\tau_0+\zeta n} r_{a,i}(t_n)$ $\forall (a,i) \in M_G$
    $T_{a,i} \leftarrow T_{a,i} + 1$ $\forall (a,i) \in M_G$
    $t_{n+1} \leftarrow t_n + \tau_0 + \zeta n$; $n \leftarrow n+1$
  end while
end procedure

In order to characterize the regret of the MG-EUCB algorithm, Proposition 1 implies that it is sufficient to bound the expected number of epochs in which our algorithm selects each sub-optimal edge. The following theorem presents an upper bound on this quantity.

Theorem 2. *Consider a finite set of m agents $\mathcal{A}$ and incentives $\mathcal{I}$ with corresponding aperiodic, irreducible Markov chains $P_{a,i}$ for each $(a,i) \in \mathcal{P}$. Let α be the MG-EUCB algorithm with mixing time sequence $\{\tau_k\}$ where $\tau_k = \tau_0 + \zeta k$, $\tau_0 > 0$, and $\zeta > 0$. Then for every $(a,i) \in \mathcal{S}$,*

$$\mathbb{E}_{\alpha}[T_{a,i}(n)] \leq \tfrac{4m^2}{\Delta^2_{a^*,i^*}}\left(\tfrac{\rho_{a^*,i^*}}{\sqrt{\tau_0}} + \sqrt{6\log n + 4\log m}\right)^2 + 2(1 + \log(n))$$

where $(a^*, i^*) = \text{argmax}_{(a_1,i_1)\in\mathcal{P}\setminus g_1^*}\left\lceil \frac{4}{\Delta^2_{a_1,i_1}}\left(\frac{\rho_{a_1,i_1}}{\sqrt{\tau_0}} + \sqrt{6\log n + 4\log m}\right)^2\right\rceil$, and $\rho_{a,i}$ is a constant specific to edge (a,i).

The full proof of the theorem is provided can be found in the supplementary material.

Proof (sketch.) There are three key ingredients to the proof: (i) linearly increasing epoch lengths, (ii) overcoming cascading errors, and (iii) application of the Azuma-Hoeffding concentration inequality.

By increasing the epoch length linearly, MG-EUCB ensures that as the algorithm converges to the optimal

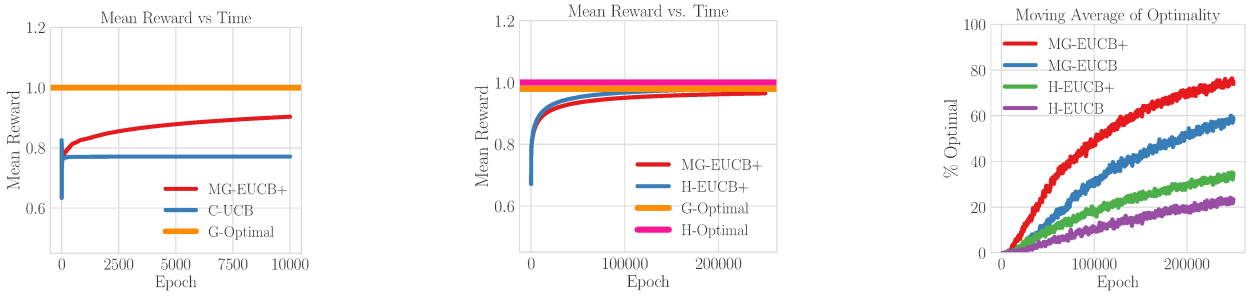


Figure 2: Synthetic Experiments: Comparison of MG-EUCB(+) and H-EUCB(+) to their respective offline solutions (G- and H-optimal, respectively) and to C-UCB (classical UCB). We use the following set up: (i) $|\mathcal{A}| = |\mathcal{I}| = |\Theta_a| = 10$ (see Supplement D for more extensive experiments) (ii) each state transition matrix $P_{a,i}$ associated with an arm $(a, i) \in \mathcal{P}$ was selected uniformly at random within the class of aperiodic and irreducible stochastic matrices; (iii) the reward for each arm, state pair $r^{\theta}_{a,i}$ is drawn i.i.d. from a distribution $\mathcal{T}_r(a, i, \theta)$ belonging to either a Bernoulli, Uniform, or Beta distribution; (iv) $\tau_0 = 50$ and $\zeta = 1$.

matching, it also plays each arm for a longer duration within an epoch. This helps the algorithm to progressively discard sub-optimal arms without selecting them too many times when the epoch length is still small. At the same time, the epoch length is long enough to allow for sufficient mixing and separation between multiple near-optimal matchings. If we fix the epoch length as a constant, the resulting regret bounds are considerably worse because the agent states may never converge to the steady-state distributions.

To address cascading errors, we provide a useful characterization. For a given (a, i), suppose that $u^k_{a,i}(t)$ refers to the average empirical reward obtained from edge (a, i) up to epoch $t-1$ plus the upper confidence bound parameter, given that edge (a, i) has been selected for exactly k times in epochs 1 to $t-1$. For any given epoch k where the algorithm selects a sub-optimal matching, i.e., $\alpha(k) \neq G^*$, we can apply Lemma 1 and get that at least one of the following conditions must be true:

1. $\mathbb{1}\{\exists j < j' | \left(u^k_{g^*_{j'}}(t) > u^k_{g^*_j}(t)\right) \wedge (g^*_{j'} \in \alpha(t))\}$
2. $\mathbb{1}\{\exists j, (a, i) \in L^*_j | \left(u^k_{g^*_j}(t) < u^k_{a,i}(t)\right) \vee ((a, i) \in \alpha(k))\} = 1$

This is a particularly useful characterization because it maps the selection of each sub-optimal edge to a familiar condition that compares the empirical rewards to the stationary rewards. Therefore, once each arm is selected for $O(\log(n))$ epochs, the empirical rewards approach the 'true' rewards and our algorithm discards sub-optimal edges. Mathematically, this can be written as

$$\mathbb{E}_\alpha[T_{a',i'}(n)] = 1 + \sum_{t=p+1}^{n} \mathbb{1}\{(a', i') \in \alpha(t)\}$$
$$\leq \ell m^2 + \sum_{j=1}^{m} \sum_{(a,i)\in L^+_j} \sum_{t=p+1}^{n} \sum_{s=1}^{t-1} \sum_{k=\ell}^{t-1} \Big(\mathbb{1}\{u^s_{g^*_j}(t) \leq u^k_{a,i}(t)\}\Big),$$

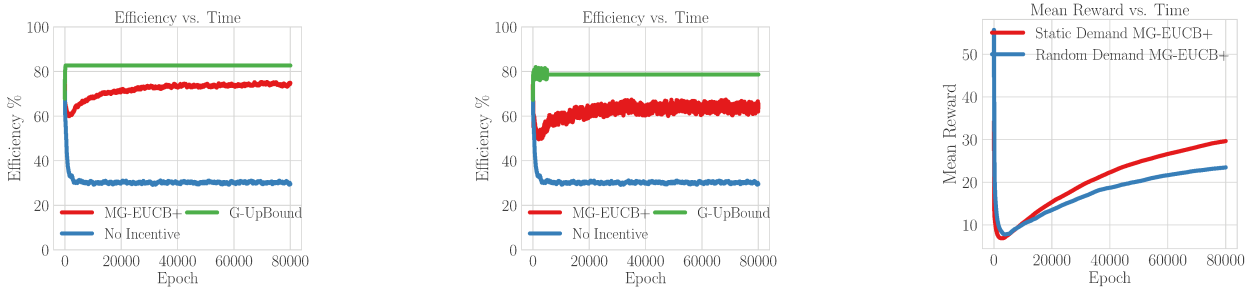
where ℓ is some carefully chosen constant, $L^+_j = L^*_j \cup \{g^*_{j+1}\}$ and $L^+_m = L^*_m$.

With this characterization, for each s, we find an upper bound on the probability of the event $\{u^s_{g^*_j}(t) \leq u^k_{a,i}(t)\}$. However, this is a non-trivial task since the reward obtained in any given epoch is *not independent* of the previous actions. Specifically, the underlying Markov process that generates the rewards is common across the edges connected to any given agent in the sense, that the initial distribution for each Markov chain that results from pulling an edge is the distribution at the end of the preceding pull. Therefore, we employ Azuma-Hoeffding (Azuma, 1967; Hoeffding, 1963), a concentration inequality that does not require independence in the arm-based observed rewards. Moreover, unlike the classical UCB analysis, the empirical reward can differ from the expected stationary reward due to the distributions $\mathcal{T}_r(a, i, \theta)$ and $\beta^k_{a,i} \neq \pi_{a,i}$. To account for this additional error term, we use bounds on the convergence rates of Markov chains to guide the choice of the confidence parameter $c^k_{a,i}(t)$ in Algorithm 2. Applying the Azuma-Hoeffding inequality, we can show that with high probability, the difference between the empirical reward and the stationary reward of edge (a, i) is no larger than $c^k_{a,i}(t)$. ■

As a direct consequence of Proposition 1 and Theorem 2, we get that for a fixed instance, the regret only increases logarithmically with n.

5 EXPERIMENTS

In this section, we present a set of illustrative experiments with our algorithm (MG-EUCB) on synthetic and



(a) Static Demand (b) Random Demand (c)

Figure 3: Bike-share Experiments: Figures 3a and 3b compare the efficiency (percentage of demand satisfied) of the bike-share system with two demand models under incentive matchings selected by MG-EUCB+ with upper and lower bounds given by the system performance when the incentives are computed via the benchmark greedy matching that uses the state information and when no incentives are offered respectively. In Figure 3c we plot the mean reward of the MG-EUCB+ algorithm with static and random demand which gives the expected number of agents who accept an incentive within each epoch.

real data. We observe much faster convergence with the greedy matching as compared to the Hungarian algorithm. Moreover, as is typical in the bandit literature (e.g., (Auer et al., 2002)), we show that a *tuned* version of our algorithm (MG-EUCB+), in which we reduce the coefficient on the $\log(n)$ term in the UCB 'confidence parameter' from six to three, further improves the convergence of our algorithm. Finally we show that our algorithm can be effectively used as an incentive design scheme to improve the performance of a bike-share system.

5.1 SYNTHETIC EXPERIMENTS

We first highlight the failure of classical UCB approaches (C-UCB)—e.g., as in (Gai et al., 2011)—for problems with correlated reward evolution. In Figure 2a, we demonstrate that C-UCB converges almost immediately to a suboptimal solution, while this is not the case for our algorithm (MG-EUCB+). In Figure 2b, we compare MG-EUCB and MG-EUCB+ with a variant of Algorithm 2 that uses the Hungarian method (H-EUCB) for matchings. While H-EUCB does have a 'marginally' higher mean reward, Figure 2c reveals that the MG-EUCB and MG-EUCB+ algorithms converge much faster to the optimum solution of the greedy matching than the Hungarian alternatives.

5.2 BIKE-SHARE EXPERIMENTS

In this problem, we seek to incentivize participants in a bike-sharing system; our goal is to alter their intended destination in order to balance the spatial supply of available bikes appropriately and meet future user demand. We use data from the Boston-based bike-sharing service Hubway (hub) to construct the example. Formally, we consider matching each agent a to an incentive $i = s'_a$, meaning the algorithm proposes that agent a travel to station s'_a as opposed to its intended destination s_a (potentially, for some monetary benefit). The agent's state θ_a controls the probability of accepting the incentive by means of a distance threshold parameter and a parameter of a Bernouilli distribution, both of which are drawn uniformly at random. More details on the data and problem setup can be found in Section D of the supplementary material.

Our bike-share simulations presented in Figure 3 show approximately a 40% improvement in system performance when compared to an environment without incentives and convergence towards an upper bound on system performance. Moreover, our algorithm achieves this significant performance increase while on average matching less than 1% of users in the system to an incentive.

6 Conclusion

We combine ideas from greedy matching, the UCB multi-armed bandit strategy, and the theory of Markov chain mixing times to propose a bandit algorithm for matching incentives to users, whose preferences are unknown a priori and evolving dynamically in time, in a resource constrained environment. For this algorithm, we derive logarithmic gap-dependent regret bounds despite the additional technical challenges of cascading sub-optimality and correlated convergence. Finally, we demonstrate the empirical performance via examples.

Acknowledgments

This work is supported by NSF Awards CNS-1736582 and CNS-1656689. T. Fiez was also supported in part by an NDSEG Fellowship.

References

Hubway: Metro-boston's bikeshare program. [available online: https://thehubway.com].

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, May 2002. doi: 10.1023/A:1013689704352.

M. G. Azar, A. Lazaric, and E. Brunskill. Regret bounds for reinforcement learning with policy advice. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 97–112, 2013.

K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J.*, 19(3):357–367, 1967. doi: 10.2748/tmj/1178243286.

A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. In *Proc. 54th Annual IEEE Symp. Foundations of Computer Science*, pages 207–216, 2013.

W. Chen, Y. Wang, Y. Yuan, and Q. Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *J. Machine Learning Research*, 17:50:1–50:33, 2016. URL http://jmlr.org/papers/v17/14-298.html.

J. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible markov chains, with an application to the exclusion process. *Ann. Appl. Probab.*, 1(1):62–87, 1991.

G. Folland. *Real Analysis*. Wiley, 2nd edition, 2007.

Y. Gai, B. Krishnamachari, and M. Liu. On the combinatorial multi-armed bandit problem with markovian rewards. In *Proc. Global Communications Conf.*, pages 1–6, 2011. doi: 10.1109/GLOCOM.2011.6134244.

M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. ISBN 0-7167-1044-7.

A. Ghosh and P. Hummel. Learning and incentives in user-generated content: multi-armed bandits with endogenous arms. In *Proc. of ITCS 2013*, pages 233–246, 2013.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. American Statistical Association*, 58(301):13–30, 1963. doi: 10.2307/2282952.

Nicole Immorlica, Gregory Stoddard, and Vasilis Syrgkanis. Social status and badge design. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 473–483, 2015.

S. Jain, B. Narayanaswamy, and Y. Narahari. A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids. In *Proc. of AAAI 2014*, pages 721–727, 2014.

T. Jaksch, R. Ortner, and P. Auer. Near-optimal Regret Bounds for Reinforcement Learning. *J. Machine Learning Research*, 11:1563–1600, 2010.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1-2):83–97, 1955.

B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proc. of UAI 2014*, pages 420–429, 2014.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. 19th Intern. Conf. World Wide Web*, pages 661–670, 2010.

T. Lu, D. Pál, and M. Pál. Contextual multi-armed bandits. In *Proc. of AISTATS 2010*, pages 485–492, 2010.

E. Mazumdar, R. Dong, V. Rúbies Royo, C. Tomlin, and S. S. Sastry. A Multi-Armed Bandit Approach for Online Expert Selection in Markov Decision Processes. *arxiv:1707.05714*, 2017.

A. Mehta and V. Mirrokni. Online ad serving: Theory and practice, 2011.

L. J. Ratliff, S. Sekar, L. Zheng, and T. Fiez. Incentives in the dark: Multi-armed bandits for evolving users with unknown type. *arxiv*, 2018.

Amir Sani, Alessandro Lazaric, and Rémi Munos. Risk-aversion in multi-armed bandits. In *Proc. of NIPS 2012*, pages 3284–3292, 2012.

S. L. Scott. Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, 31(1):37–45, 2015.

A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and Andreas Krause. Incentivizing users for balancing bike sharing systems. In *Proc. of AAAI 2015*, pages 723–729, 2015.

Cem Tekin and Mingyan Liu. Online algorithms for the multi-armed bandit problem with markovian rewards.

In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1675–1682. IEEE, 2010.

Cem Tekin and Mingyan Liu. Online Learning of Rested and Restless Bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.

L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artif. Intell.*, 214: 89–111, 2014.

Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 1113–1122, 2015.