Matroid Bandits: Fast Combinatorial Optimization with Learning

Branislav Kveton, **Zheng Wen**, **Azin Ashkan**, **Hoda Eydgahi**, and **Brian Eriksson**
Technicolor Labs
Los Altos, CA
$\{$*branislav.kveton,zheng.wen,azin.ashkan,hoda.eydgahi,brian.eriksson*$\}$*@technicolor.com*

Abstract

A matroid is a notion of independence in combinatorial optimization which is closely related to computational efficiency. In particular, it is well known that the maximum of a constrained modular function can be found greedily if and only if the constraints are associated with a matroid. In this paper, we bring together the ideas of bandits and matroids, and propose a new class of combinatorial bandits, *matroid bandits*. The objective in these problems is to learn how to maximize a modular function on a matroid. This function is stochastic and initially unknown. We propose a practical algorithm for solving our problem, *Optimistic Matroid Maximization (*`OMM`*)*; and prove two upper bounds, gap-dependent and gap-free, on its regret. Both bounds are sublinear in time and at most linear in all other quantities of interest. The gap-dependent upper bound is tight and we prove a matching lower bound on a partition matroid bandit. Finally, we evaluate our method on three real-world problems and show that it is practical.

1 Introduction

Combinatorial optimization is a well-established field that has many practical applications, ranging from resource allocation [14] to designing network routing protocols [20]. Modern combinatorial optimization problems are often so massive that even low-order polynomial-time solutions are not practical. Fortunately, many important problems, such as finding a minimum spanning tree, can be solved greedily. Such problems can be often viewed as optimization on a *matroid* [25], a notion of independence in combinatorial optimization which is closely related to computational efficiency. In particular, it is well known that the maximum of a constrained modular function can be found greedily if and only if all feasible solutions to the problem are the independent sets of a matroid [8]. Matroids are common in practice because they generalize many notions of independence, such as linear independence and forests in graphs.

In this paper, we propose an algorithm for learning how to maximize a stochastic modular function on a matroid. The modular function is represented as the sum of the weights of up to K items, which are chosen from the ground set E of a matroid, which has L items. The weights of the items are stochastic and represented as a vector $\mathbf{w} \in [0, 1]^L$. The vector $\mathbf{w}$ is drawn i.i.d. from a probability distribution P. The distribution P is initially unknown and we learn it by interacting repeatedly with the environment.

Many real-world optimization problems can be formulated in our setting, such as building a spanning tree for network routing [20]. When the delays on the links of the network are stochastic and their distribution is known, this problem can be solved by finding a minimum spanning tree. When the distribution is unknown, it must be learned, perhaps by exploring routing networks that seem initially suboptimal. We return to this problem in our experiments.

This paper makes three main contributions. First, we bring together the concepts of matroids [25] and bandits [15, 3], and propose a new class of combinatorial bandits, *matroid bandits*. On one hand, matroid bandits can be viewed as a new learning framework for a broad and important class of combinatorial optimization problems. On the other hand, matroid bandits are a class of K-step bandit problems that can be solved both computationally and sample efficiently.

Second, we propose a simple greedy algorithm for solving our problem, which explores based on the optimism in the face of uncertainty. We refer to our approach as *Optimistic Matroid Maximization (*`OMM`*)*. `OMM` is both computationally and sample efficient. In particular, the time complexity of `OMM` is $O(L \log L)$ per episode, comparable to that of sorting L numbers. Moreover, the expected cumulative regret of `OMM` is sublinear in the number of episodes, and at most linear in the number of items L and the maximum number of chosen items K.

Finally, we evaluate `OMM` on three real-world problems. In the first problem, we learn routing networks. In the second problem, we learn a policy for assigning loans in a microfinance network that maximizes chances that the loans are repaid. In the third problem, we learn a movie recommendation policy. All three problems can be solved efficiently in our framework. This demonstrates that OMM is practical and can solve a wide range of real-world problems.

We adopt the following notation. We write $A + e$ instead of $A \cup \{e\}$, and $A + B$ instead of $A \cup B$. We also write $A - e$ instead of $A \setminus \{e\}$, and $A - B$ instead of $A \setminus B$.

2 Matroids

A *matroid* is a pair $M = (E, \mathcal{I})$, where $E = \{1, \dots, L\}$ is a set of L items, called the *ground set*, and $\mathcal{I}$ is a family of subsets of E, called the *independent sets*. The family $\mathcal{I}$ is defined by the following properties. First, $\emptyset \in \mathcal{I}$. Second, every subset of an independent set is independent. Finally, for any $X \in \mathcal{I}$ and $Y \in \mathcal{I}$ such that $|X| = |Y| + 1$, there must exist an item $e \in X - Y$ such that $Y + e \in \mathcal{I}$. This is known as the *augmentation property*. We denote by:

$$E(X) = \{e : e \in E - X,\ X + e \in \mathcal{I}\} \tag{1}$$

the set of items that can be added to set X such that the set remains independent.

A set is a *basis* of a matroid if it is a maximal independent set. All bases of a matroid have the same cardinality [25], which is known as the *rank* of a matroid. In this work, we denote the rank by K.

A *weighted matroid* is a matroid associated with a vector of non-negative weights $\mathbf{w} \in (\mathbb{R}^+)^L$. The e-th entry of $\mathbf{w}$, $\mathbf{w}(e)$, is the weight of item e. We denote by:

$$f(A, \mathbf{w}) = \sum_{e \in A} \mathbf{w}(e) \tag{2}$$

the sum of the weights of all items in set A. The problem of finding a *maximum-weight basis* of a matroid:

$$A^* = \arg\max_{A \in \mathcal{I}} f(A, \mathbf{w}) = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \mathbf{w}(e) \tag{3}$$

is a well-known combinatorial optimization problem. This problem can be solved greedily (Algorithm 1). The greedy algorithm has two main stages. First, A^* is initialized to $\emptyset$. Second, all items in the ground set are sorted according to their weights, from the highest to the lowest, and greedily added to A^* in this order. The item is added to the set A^* only if it does not make the set dependent.

Algorithm 1 The greedy method for finding a maximum-weight basis of a matroid [8].

```
Input: Matroid $M = (E, \mathcal{I})$, weights $\mathbf{w}$

$A^* \leftarrow \emptyset$
Let $e_1, \dots, e_L$ be an ordering of items such that:
  $\mathbf{w}(e_1) \geq \dots \geq \mathbf{w}(e_L)$
for all $i = 1, \dots, L$ do
  if $(e_i \in E(A^*))$ then
    $A^* \leftarrow A^* + e_i$
  end if
end for
```

3 Matroid Bandits

A *minimum spanning tree* is a maximum-weight basis of a matroid. The ground set E of this matroid are the edges of a graph. A set of edges is considered to be independent if it does not contain a cycle. Each edge e is associated with a weight $\mathbf{w}(e) = u_{\max} - \mathbf{u}(e)$, where $u_{\max} = \max_e \mathbf{u}(e)$ and $\mathbf{u}(e)$ is the weight of edge e in the original graph.

The minimum spanning tree cannot be computed when the weights $\mathbf{w}(e)$ of the edges are unknown. This may happen in practice. For instance, consider the problem of building a routing network, which is represented as a spanning tree, where the expected delays on the links of the network are initially unknown. In this work, we study a variant of maximizing a modular function on a matroid that can address this kind of problems.

3.1 Model

We formalize our learning problem as a matroid bandit. A *matroid bandit* is a pair (M, P), where M is a matroid and P is a probability distribution over the weights $\mathbf{w} \in \mathbb{R}^L$ of items E in M. The e-th entry of $\mathbf{w}$, $\mathbf{w}(e)$, is the weight of item e. The weights $\mathbf{w}$ are stochastic and drawn i.i.d. from the distribution P. We denote the expected weights of the items by $\bar{\mathbf{w}} = \mathbb{E}[\mathbf{w}]$ and assume that each of these weights is non-negative, $\bar{\mathbf{w}}(e) \geq 0$ for all $e \in E$.

Each item e is associated with an *arm* and we assume that *multiple arms* can be pulled. A subset of arms $A \subseteq E$ can be pulled if and only if A is an independent set. The return for pulling arms A is $f(A, \mathbf{w})$ (Equation 2), the sum of the weights of all items in A. After the arms A are pulled, we observe the weight of each item in A, $\mathbf{w}(e)$ for all $e \in A$. This model of feedback is known as *semi-bandit* [2].

We assume that the matroid M is known and that the distribution P is unknown. Without loss of generality, we assume that the support of P is a bounded subset of $[0, 1]^L$. We would like to stress that we do not make any structural assumptions on P.

The optimal solution to our problem is a maximum-weight basis in expectation:

$$A^* = \arg\max_{A \in \mathcal{I}} \mathbb{E}_{\mathbf{w}}[f(A, \mathbf{w})] = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \bar{\mathbf{w}}(e). \tag{4}$$

Algorithm 2 `OMM`: Optimistic matroid maximization.

```
Input: Matroid $M = (E, \mathcal{I})$

// Initialization
Observe $\mathbf{w}_0 \sim P$
$\hat{w}_{e,1} \leftarrow \mathbf{w}_0(e)$                              $\forall e \in E$
$T_e(0) \leftarrow 1$                                        $\forall e \in E$

for all $t = 1, \dots, n$ do
  // Compute UCBs
  $U_t(e) \leftarrow \hat{w}_{e,T_e(t-1)} + c_{t-1,T_e(t-1)}$            $\forall e \in E$

  // Find a maximum-weight basis with respect to $U_t$
  $A^t \leftarrow \emptyset$
  Let $e_1^t, \dots, e_L^t$ be an ordering of items such that:
    $U_t(e_1^t) \geq \dots \geq U_t(e_L^t)$
  for all $i = 1, \dots, L$ do
    if ($e_i^t \in E(A^t)$) then
      $A^t \leftarrow A^t + e_i^t$
    end if
  end for
  Observe $\{\mathbf{w}_t(e) : e \in A^t\}$, where $\mathbf{w}_t \sim P$

  // Update statistics
  $T_e(t) \leftarrow T_e(t-1)$                                 $\forall e \in E$
  $T_e(t) \leftarrow T_e(t) + 1$                               $\forall e \in A^t$
  $\hat{w}_{e,T_e(t)} \leftarrow \dfrac{T_e(t-1)\hat{w}_{e,T_e(t-1)} + \mathbf{w}_t(e)}{T_e(t)}$   $\forall e \in A^t$
end for
```

The above optimization problem is equivalent to the problem in Equation 3. Therefore, it can be solved greedily by Algorithm 1 when the expected weights $\bar{\mathbf{w}}$ are known.

Our learning problem is *episodic*. In episode t, we choose a basis A^t and gain $f(A^t, \mathbf{w}_t)$, where $\mathbf{w}_t$ is the realization of the stochastic weights in episode t. Our goal is to learn a policy, a sequence of bases, that minimizes the *expected cumulative regret* in n episodes:

$$R(n) = \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[\sum_{t=1}^{n} R_t(\mathbf{w}_t) \right], \qquad (5)$$

where $R_t(\mathbf{w}_t) = f(A^*, \mathbf{w}_t) - f(A^t, \mathbf{w}_t)$ is the regret in episode t.

3.2 Algorithm

Our solution is designed based on the *optimism in the face of uncertainty* principle [17]. In particular, it is a variant of the greedy method for finding a maximum-weight basis of a matroid where the expected weight $\bar{\mathbf{w}}(e)$ of each item e is substituted with its optimistic estimate $U_t(e)$. Therefore, we refer to our approach as *Optimistic Matroid Maximization (*`OMM`*)*.

The pseudocode of our algorithm is given in Algorithm 2. The algorithm can be summarized as follows. First, at the beginning of each episode t, we compute the *upper confidence bound (UCB)* on the weight of each item e:

$$U_t(e) = \hat{w}_{e,T_e(t-1)} + c_{t-1,T_e(t-1)}, \qquad (6)$$

where $\hat{w}_{e,T_e(t-1)}$ is our estimate of $\bar{\mathbf{w}}(e)$ at the beginning of episode t, $c_{t-1,T_e(t-1)}$ represents the radius of the confidence interval around this estimate, and $T_e(t-1)$ is the number of times that `OMM` chooses item e before episode t. Second, we order all items e by their UCBs (Equation 6), from the highest to the lowest, and then add them greedily to A^t in this order. The item is added to the set A^t only if it does not make the set dependent. Finally, we choose the basis A^t, observe the weights of all items in the basis, and update our model $\hat{w}$ of the world.

The radius:

$$c_{t,s} = \sqrt{2 \log(t)/s} \qquad (7)$$

is defined such that each upper confidence bound $U_t(e)$ is with high probability an upper bound on the weight $\bar{\mathbf{w}}(e)$. The role of the UCBs is to encourage exploration of items that are not chosen very often. As the number of episodes increases, the estimates of the weights $\bar{\mathbf{w}}$ improve and `OMM` starts exploiting best items. The $\log(t)$ term increases with time t and enforces exploration, to avoid linear regret.

`OMM` is a greedy algorithm and therefore is extremely computationally efficient. In particular, let the time complexity of checking for independence, $e_i^t \in E(A^t)$, be $O(g(|A^t|))$. Then the time complexity of `OMM` is $O(L(\log L + g(K)))$ per episode, comparable to that of sorting L numbers. The design of our algorithm is not surprising and is motivated by prior work [12]. The main challenge is to derive a tight upper bound on the regret of `OMM`, which would reflect the structure of our problem.

4 Analysis

In this section, we analyze the regret of `OMM`. Our analysis is organized as follows. First, we introduce basic concepts and notation. Second, we show how to decompose the regret of `OMM` in a single episode. In particular, we partition the regret of a suboptimal basis into the sum of the regrets of individual items. This part of the analysis relies heavily on the structure of a matroid and is the most novel. Third, we derive two upper bounds, gap-dependent and gap-free, on the regret of `OMM`. Fourth, we prove a lower bound that matches the gap-dependent upper bound. Finally, we summarize the results of our analysis.

4.1 Notation

Before we present our results, we introduce notation used in our analysis. The *optimal basis* is $A^* = \{a_1^*, \dots, a_K^*\}$.

We assume that the items in A^* are ordered such that a^*_k is the k-th item with the highest expected weight. In episode t, `OMM` chooses a basis $A^t = \{a^t_1, \dots, a^t_K\}$, where a^t_k is the k-th item chosen by `OMM`. We say that item e is *suboptimal* if it belongs to $\bar{A}^* = E - A^*$, the *set of suboptimal items*. For any pair of suboptimal and optimal items, $e \in \bar{A}^*$ and a^*_k, we define a *gap*:

$$\Delta_{e,k} = \bar{\mathbf{w}}(a^*_k) - \bar{\mathbf{w}}(e) \tag{8}$$

and use it as a measure of how difficult it is to discriminate the items. For every item $e \in \bar{A}^*$, we define a set:

$$\mathcal{O}_e = \{k : \Delta_{e,k} > 0\}, \tag{9}$$

the indices of items in A^* whose expected weight is higher than that of item e. The cardinality of $\mathcal{O}_e$ is $K_e = |\mathcal{O}_e|$.

4.2 Regret Decomposition

Our decomposition is motivated by the observation that all bases of a matroid are of the same cardinality. As a result, the difference in the expected values of any two bases can be always written as the sum of differences in the weights of their items. In particular:

$$\mathbb{E}_{\mathbf{w}}\left[f(A^*, \mathbf{w}) - f(A^t, \mathbf{w})\right] = \sum_{k=1}^{K} \Delta_{a^t_k, \pi(k)}, \tag{10}$$

where $\pi : \{1, \dots, K\} \to \{1, \dots, K\}$ is an arbitrary bijection from A^t to A^* such that $\pi(k)$ is the index of the item in A^* that is paired with the k-th item in A^t. In this work, we focus on one particular bijection.

Lemma 1. *For any two matroid bases A^* and A^t, there exists a bijection $\pi : \{1, \dots, K\} \to \{1, \dots, K\}$ such that:*

$$\left\{a^t_1, \dots, a^t_{k-1}, a^*_{\pi(k)}\right\} \in \mathcal{I} \quad \forall k = 1, \dots, K.$$

In addition, $\pi(k) = i$ when $a^t_k = a^*_i$ for some i.

Proof. The lemma is proved in Appendix. ■

The bijection π in Lemma 1 has two important properties. First, $\left\{a^t_1, \dots, a^t_{k-1}, a^*_{\pi(k)}\right\} \in \mathcal{I}$ for all k. In other words, `OMM` can choose item $a^*_{\pi(k)}$ at step k. However, `OMM` selects item a^t_k. By the design of `OMM`, this can happen only when the UCB of item a^t_k is larger or equal to that of item $a^*_{\pi(k)}$. As a result, we know that $U_t(a^t_k) \geq U_t(a^*_{\pi(k)})$ in all steps k. Second, Lemma 1 guarantees that every optimal item in A^t is paired with the same item in A^*.

In the rest of the paper, we represent the bijection π using an indicator function. The indicator function:

$$\mathbb{1}_{e,k}(t) = \mathbb{1}\{\exists i : a^t_i = e,\ \pi(i) = k\} \tag{11}$$

indicates the event that item e is chosen instead of item a^*_k in episode t. Based on our new representation, we rewrite Equation 10 as:

$$\begin{aligned} \sum_{k=1}^{K} \Delta_{a^t_k, \pi(k)} &= \sum_{e \in \bar{A}^*} \sum_{k=1}^{K} \Delta_{e,k} \mathbb{1}_{e,k}(t) \\ &\leq \sum_{e \in \bar{A}^*} \sum_{k=1}^{K_e} \Delta_{e,k} \mathbb{1}_{e,k}(t) \end{aligned} \tag{12}$$

and then bound it from above. The last inequality is due to neglecting the negative gaps.

The above analysis applies to any basis A^t in any episode t. The results of our analysis are summarized below.

Theorem 1. *The expected regret of choosing any basis A^t in episode t is bounded as:*

$$\mathbb{E}_{\mathbf{w}}\left[f(A^*, \mathbf{w}) - f(A^t, \mathbf{w})\right] \leq \sum_{e \in \bar{A}^*} \sum_{k=1}^{K_e} \Delta_{e,k} \mathbb{1}_{e,k}(t).$$

The indicator function $\mathbb{1}_{e,k}(t)$ indicates the event that item e is chosen instead of item a^*_k in episode t. When the event $\mathbb{1}_{e,k}(t)$ happens, $U_t(e) \geq U_t(a^*_k)$. Moreover:

$$\begin{aligned} \sum_{e \in \bar{A}^*} \sum_{k=1}^{K_e} \mathbb{1}_{e,k}(t) &\leq K & \forall t \\ \sum_{k=1}^{K_e} \mathbb{1}_{e,k}(t) &\leq 1 & \forall t, e \in \bar{A}^*. \end{aligned}$$

The last two inequalities follow from the fact that $\mathbb{1}_{e,k}(t)$ is a bijection from A^t to A^*, every item in the suboptimal basis A^t is matched with one unique item in A^*.

One remarkable aspect of our regret decomposition is that the exact form of the bijection is not required in the rest of our analysis. We only rely on the properties of $\mathbb{1}_{e,k}(t)$ that are stated in Theorem 1.

4.3 Upper Bounds

Our first result is a gap-dependent bound.

Theorem 2 (gap-dependent bound)**.** *The expected cumulative regret of* `OMM` *is bounded as:*

$$R(n) \leq \sum_{e \in \bar{A}^*} \frac{16}{\Delta_{e,K_e}} \log n + \sum_{e \in \bar{A}^*} \sum_{k=1}^{K_e} \Delta_{e,k} \frac{4}{3}\pi^2.$$

Proof. First, we bound the expected regret in episode t using Theorem 1:

$$\begin{aligned} R(n) &= \sum_{t=1}^{n} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_{t-1}}[\mathbb{E}_{\mathbf{w}_t}[R_t(\mathbf{w}_t)]] \\ &\le \sum_{t=1}^{n} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_{t-1}}\left[\sum_{e \in \bar{A}^*}\sum_{k=1}^{K_e} \Delta_{e,k}\mathbb{1}_{e,k}(t)\right] \\ &= \sum_{e \in \bar{A}^*}\sum_{k=1}^{K_e} \Delta_{e,k}\mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\mathbb{1}_{e,k}(t)\right]. \qquad (13) \end{aligned}$$

Second, we bound the expected cumulative regret associated with each item $e \in \bar{A}^*$. The key idea of this step is to decompose the indicator $\mathbb{1}_{e,k}(t)$ as:

$$\begin{aligned} \mathbb{1}_{e,k}(t) = \mathbb{1}_{e,k}(t)\mathbb{1}\{T_e(t-1) \le \ell_{e,k}\} + \qquad (14) \\ \mathbb{1}_{e,k}(t)\mathbb{1}\{T_e(t-1) > \ell_{e,k}\} \end{aligned}$$

and choose $\ell_{e,k}$ appropriately. By Lemma 2, the regret associated with $T_e(t-1) > \ell_{e,k}$ is bounded as:

$$\begin{aligned} &\sum_{k=1}^{K_e} \Delta_{e,k}\mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\mathbb{1}_{e,k}(t)\mathbb{1}\{T_e(t-1) > \ell_{e,k}\}\right] \\ &\quad \le \sum_{k=1}^{K_e} \Delta_{e,k}\frac{4}{3}\pi^2 \qquad (15) \end{aligned}$$

when $\ell_{e,k} = \left\lfloor \frac{8}{\Delta_{e,k}^2}\log n \right\rfloor$. For the same value of $\ell_{e,k}$, the regret associated with $T_e(t-1) \le \ell_{e,k}$ is bounded as:

$$\begin{aligned} &\sum_{k=1}^{K_e} \Delta_{e,k}\mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\mathbb{1}_{e,k}(t)\mathbb{1}\{T_e(t-1) \le \ell_{e,k}\}\right] \\ &\quad \le \max_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\sum_{k=1}^{K_e}\Delta_{e,k}\mathbb{1}_{e,k}(t) \times \right. \qquad (16) \\ &\qquad\qquad \left. \mathbb{1}\left\{T_e(t-1) \le \frac{8}{\Delta_{e,k}^2}\log n\right\}\right]. \end{aligned}$$

The next step of our proof is based on three observations. First, the gaps are ordered such that $\Delta_{e,1} \ge \ldots \ge \Delta_{e,K_e}$. Second, by the design of OMM, the counter $T_e(t)$ increases when the event $\mathbb{1}_{e,k}(t)$ happens, for any k. Finally, by Theorem 1, $\sum_{k=1}^{K_e}\mathbb{1}_{e,k}(t) \le 1$ for any given e and t. Based on these facts, we bound Equation 16 from above by:

$$\left[\Delta_{e,1}\frac{1}{\Delta_{e,1}^2} + \sum_{k=2}^{K_e}\Delta_{e,k}\left(\frac{1}{\Delta_{e,k}^2} - \frac{1}{\Delta_{e,k-1}^2}\right)\right] 8\log n. \qquad (17)$$

By Lemma 3, the above term is bounded by $\dfrac{16}{\Delta_{e,K_e}}\log n$. Finally, we combine all of the above inequalities and get:

$$\begin{aligned} &\sum_{k=1}^{K_e} \Delta_{e,k}\mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\mathbb{1}_{e,k}(t)\right] \\ &\quad \le \frac{16}{\Delta_{e,K_e}}\log n + \sum_{k=1}^{K_e}\Delta_{e,k}\frac{4}{3}\pi^2. \qquad (18) \end{aligned}$$

Our main claim is obtained by summing over all suboptimal items $e \in \bar{A}^*$. ■

We also prove a gap-free bound.

Theorem 3 (gap-free bound)**.** *The expected cumulative regret of* OMM *is bounded as:*

$$R(n) \le 8\sqrt{KLn\log n} + \frac{4}{3}\pi^2 KL.$$

Proof. The key idea is to decompose the expected cumulative regret of OMM into two parts, where the gaps are larger than ε and at most ε. We analyze each part separately and then set ε to get the desired result.

Let $K_{e,\varepsilon}$ be the number of optimal items whose expected weight is higher than that of item e by more than ε and:

$$Z_{e,k}(n) = \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\mathbb{1}_{e,k}(t)\right]. \qquad (19)$$

Then, based on Equation 13, the regret of OMM is bounded for any ε as:

$$\begin{aligned} R(n) \le \sum_{e \in \bar{A}^*}\sum_{k=1}^{K_{e,\varepsilon}}\Delta_{e,k}Z_{e,k}(n) + \qquad (20) \\ \sum_{e \in \bar{A}^*}\sum_{k=K_{e,\varepsilon}+1}^{K_e}\Delta_{e,k}Z_{e,k}(n). \end{aligned}$$

The first term can be bounded similarly to Equation 18:

$$\begin{aligned} &\sum_{e \in \bar{A}^*}\sum_{k=1}^{K_{e,\varepsilon}}\Delta_{e,k}Z_{e,k}(n) \\ &\quad \le \sum_{e \in \bar{A}^*}\frac{16}{\Delta_{e,K_{e,\varepsilon}}}\log n + \sum_{e \in \bar{A}^*}\sum_{k=1}^{K_{e,\varepsilon}}\Delta_{e,k}\frac{4}{3}\pi^2 \\ &\quad \le \frac{16}{\varepsilon}L\log n + \frac{4}{3}\pi^2 KL. \qquad (21) \end{aligned}$$

The second term is bounded trivially as:

$$\sum_{e \in \bar{A}^*}\sum_{k=K_{e,\varepsilon}+1}^{K_e}\Delta_{e,k}Z_{e,k}(n) \le \varepsilon Kn \qquad (22)$$

because all gaps $\Delta_{e,k}$ are bounded by ε and the maximum number of suboptimally chosen items in n episodes is Kn (Theorem 1). Based on our upper bounds, we get:

$$R(n) \le \frac{16}{\varepsilon}L\log n + \varepsilon Kn + \frac{4}{3}\pi^2 KL \qquad (23)$$

and then set $\varepsilon = 4\sqrt{\dfrac{L\log n}{Kn}}$. This concludes our proof. ■

4.4 Lower Bounds

We derive an asymptotic lower bound on the expected cumulative regret $R(n)$ that has the same dependence on the gap and n as the upper bound in Theorem 2. This bound is proved on a class of matroid bandits that are equivalent to K Bernoulli bandits.

Specifically, we prove the lower bound on a *partition matroid bandit*, which is defined as follows. Let E be a set of L items and $B_1, \dots, B_K$ be a partition of this set. Let the family of independent sets be defined as:

$$\mathcal{I} = \left\{ I \subseteq E : (\forall k : |I \cap B_k| \leq 1) \right\}. \tag{24}$$

Then $M = (E, \mathcal{I})$ is a *partition matroid* of rank K. Let P be a probability distribution over the weights of the items, where the weight of each item is distributed independently of the other items. Let the weight of item e be drawn i.i.d. from a Bernoulli distribution with mean:

$$\bar{\mathbf{w}}(e) = \begin{cases} 0.5 & e = \min_{i \in B_k} i \\ 0.5 - \Delta & \text{otherwise,} \end{cases} \tag{25}$$

where $\Delta > 0$. Then $\tilde{B} = (M, P)$ is our partition matroid bandit. The key property of $\tilde{B}$ is that it is equivalent to K independent Bernoulli bandits, one for each partition. The optimal item in each partition is the item with the smallest index, $\min_{i \in B_k} i$. All gaps are Δ.

To formalize our result, we need to introduce the notion of *consistent algorithms*. We say that the algorithm is consistent if for any matroid bandit, any suboptimal $e \in \bar{A}^*$, and any $\alpha > 0$, $\mathbb{E}[T_e(n)] = o(n^\alpha)$, where $T_e(n)$ is the number of times that item e is chosen in n episodes. In the rest of our analysis, we focus only on consistent algorithms. This is without loss of generality. In particular, by definition, an inconsistent algorithm performs poorly on some problems, and therefore extremely well on others. Because of this, it is difficult to prove good problem-dependent lower bounds for inconsistent algorithms. Our main claim is below.

Theorem 4. *For any partition matroid bandit $\tilde{B}$ that is defined in Equations 24 and 25, and parameterized by L, K, and $0 < \Delta < 0.5$; the regret of any consistent algorithm is bounded from below as:*

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geq \frac{L - K}{4\Delta}.$$

Proof. The theorem is proved as follows:

$$\begin{aligned} \liminf_{n \to \infty} \frac{R(n)}{\log n} &\geq \sum_{k=1}^{K} \sum_{e \in B_k - A^*} \frac{\Delta}{\mathrm{kl}(0.5 - \Delta, 0.5)} \\ &= \frac{(L - K)\Delta}{\mathrm{kl}(0.5 - \Delta, 0.5)} \\ &\geq \frac{L - K}{4\Delta}, \end{aligned} \tag{26}$$

where $\mathrm{kl}(0.5 - \Delta, 0.5)$ is the KL divergence between two Bernoulli variables with means $0.5 - \Delta$ and 0.5. The first inequality is due to Theorem 2.2 [4], which is applied separately to each partition B_k. The second inequality is due to $\mathrm{kl}(p, q) \leq \frac{(p-q)^2}{q(1-q)}$, where $p = 0.5 - \Delta$ and $q = 0.5$. ■

4.5 Summary of Theoretical Results

We prove two upper bounds on the regret of `OMM`, one gap-dependent and one gap-free. These bounds can be summarized as:

$$\begin{array}{ll} \text{Theorem 2} & O(L(1/\Delta) \log n) \\ \text{Theorem 3} & O(\sqrt{KLn \log n}), \end{array} \tag{27}$$

where $\Delta = \min_e \min_{k \in \mathcal{O}_e} \Delta_{e,k}$. Both bounds are sublinear in the number of episodes n, and at most linear in the rank K of the matroid and the number of items L. In other words, they scale favorably with all quantities of interest and as a result we expect them to be practical.

Our upper bounds are reasonably tight. More specifically, the gap-dependent upper bound in Theorem 2 matches the lower bound in Theorem 4, which is proved on a partition matroid bandit. Furthermore, the gap-free upper bound in Theorem 3 matches the lower bound of Audibert *et al.* [2] in adversarial combinatorial semi-bandits, up to a factor of $\sqrt{\log n}$.

Our gap-dependent upper bound has the same form as the bound of Auer *et al.* [3] for multi-armed bandits. This observation suggests that the sample complexity of learning a maximum-weight basis of a matroid is comparable to that of the multi-armed bandit. The only major difference is in the definitions of the gaps. We conclude that learning with matroids is extremely sample efficient.

5 Experiments

Our algorithm is evaluated on three matroid bandit problems: graphic (Section 5.1), transversal (Section 5.2), and linear (Section 5.3).

All experiments are episodic. In each episode, `OMM` selects a basis A^t, observes the weights of the individual items in that basis, and then updates its model of the environment. The performance of `OMM` is measured by the *expected per-step return* in n episodes:

$$\frac{1}{n} \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[\sum_{t=1}^{n} f(A^t, \mathbf{w}_t) \right], \tag{28}$$

the expected cumulative return in n episodes divided by n. `OMM` is compared to two baselines. The first baseline is the maximum-weight basis A^* in expectation. The basis A^* is computed as in Equation 4 and is our notion of optimality.

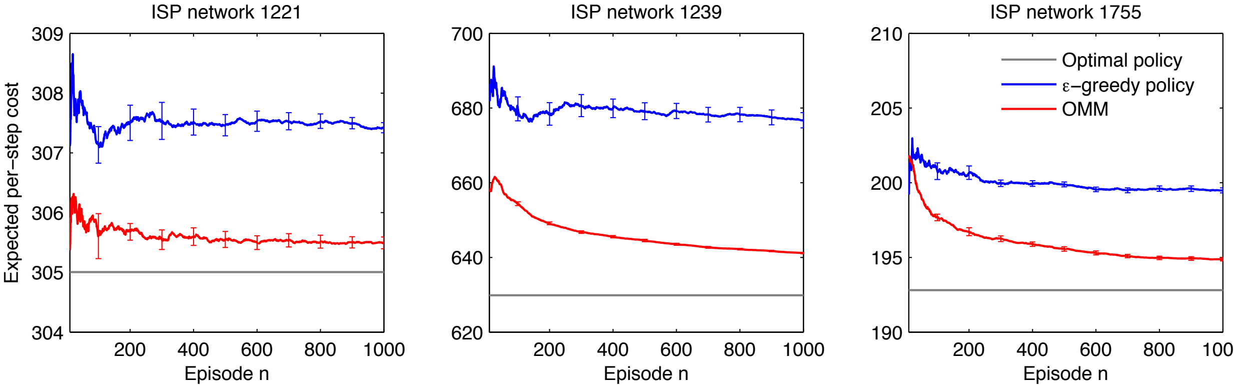


Figure 1: The expected per-step cost of building three minimum spanning trees in up to 10^3 episodes.

ISP network	Number of nodes	Number of edges	Minimum latency	Maximum latency	Average latency	Optimal policy	ε-greedy policy	`OMM`
1221	108	153	1	17	2.78	305.00	307.42 ± 0.08	305.49 ± 0.10
1239	315	972	1	64	3.20	629.88	676.74 ± 2.03	641.17 ± 0.18
1755	87	161	1	31	2.91	192.81	199.49 ± 0.16	194.88 ± 0.11
3257	161	328	1	47	4.30	550.85	570.35 ± 0.63	559.80 ± 0.10
3967	79	147	1	44	5.19	306.80	320.30 ± 0.52	308.54 ± 0.08
6461	141	374	1	45	6.32	376.27	424.78 ± 1.54	381.48 ± 0.07

Table 1: The description of six ISP networks from our experiments and the expected per-step cost of building minimum spanning trees on these networks in 10^3 episodes. All latencies and costs are in milliseconds.

The second baseline is an ε-greedy policy, where $\varepsilon = 0.1$. This setting of ε is common in practice and corresponds to 10% exploration.

5.1 Graphic Matroid

In the first experiment, we evaluate `OMM` on the problem of learning a routing network for an Internet service provider (ISP). We make the assumption that the routing network is a spanning tree. Our objective is to learn a tree that has the lowest expected latency on its edges.

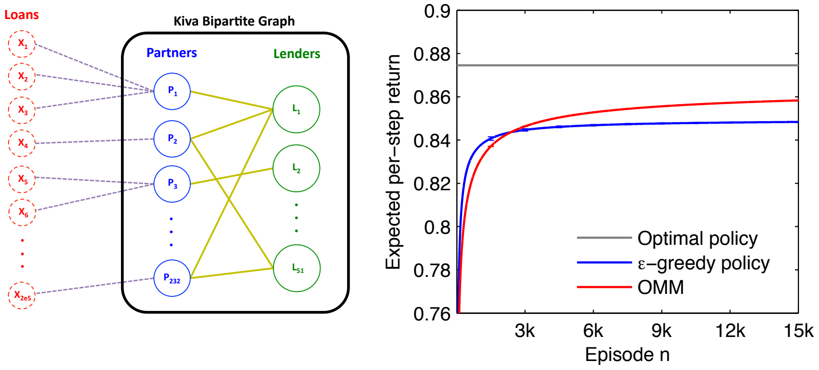
Our problem can be formulated as a *graphic matroid bandit*. The ground set E are the edges of a graph, which represents the topology of a network. We experiment with six networks from the *RocketFuel* dataset [23], which contain up to 300 nodes and 10^3 edges (Table 1). A set of edges is considered *independent* if it does not contain a cycle. The latency of edge e is $\mathbf{w}(e) = \bar{\mathbf{w}}(e) - 1 + \varepsilon$, where $\bar{\mathbf{w}}(e)$ is the expected latency, which is recorded in our dataset; and $\varepsilon \sim \mathrm{Exp}(1)$ is exponential noise. The latency $\bar{\mathbf{w}}(e)$ ranges from one to 64 milliseconds. Our noise model is motivated by the following observation. The latency in ISP networks can be mostly explained by geographical distances [7], the expected latency $\bar{\mathbf{w}}(e)$. The noise tends to be small, on the order of a few hundred microseconds, and it is unlikely to cause high latency.

In Figure 1, we report our results from three ISP networks. We observe the same trends on all networks. First, the expected cost of `OMM` approaches that of the optimal solution A^* as the number of episodes increases. Second, `OMM` outperforms the ε-greedy policy in less than 10 episodes. The expected costs of all policies on all networks are reported in Table 1. We observe again that `OMM` consistently outperforms the ε-greedy policy, often by a large margin.

`OMM` learns quickly because all of our networks are sparse. In particular, the number of edges in each network is never more than four times larger than the number of edges in its spanning tree. Therefore, at least in theory, each edge can be observed at least once in four episodes and our method can learn quickly the mean latency of each edge.

5.2 Transversal Matroid

In the second experiment, we study the assignment of lending institutions (known as *partners*) to *lenders* in a microfinance setting, such as Kiva [1]. This problem can be formulated under a family of matroids, called *transversal* matroids [9]. The ground set E of a transversal matroid is the set of left vertices of the corresponding bipartite graph, and the independence set $\mathcal{I}$ consists of the sets of left vertices that belong to all possible matchings in the graph such that no two edges in a matching share an endpoint. The weight $\bar{\mathbf{w}}(e)$ is the weight associated with the left vertices of the bipartite graph. The goal is to learn a transversal of the bipartite graph that maximizes the overall weight of selected



(a)

(b)

Partner id	$\bar{\mathbf{w}}(e)$	Lender id	Num of partners	Avg rate
46	1.0	31	200	0.728
70	1.0	2	2	0.924
72	1.0	20	195	0.725
88	1.0	23	207	0.724
168	0.983	44	49	0.712
231	0.981	48	186	0.723
179	0.970	24	149	0.743
157	0.951	10	180	0.735
232	0.940	40	10	0.718
123	0.934	42	113	0.721
142	0.925	7	168	0.745
130	0.919	32	23	0.690

(c)

Figure 2: **(a)** The Kiva dataset can be modeled as a bipartite graph connecting lenders to field partners, which, in turn, fund several loans in the region. **(b)** The expected per-step return of finding maximum weight transversal in up to 15k episodes. **(c)** Top 12 selected partners assigned based on their mean success rate in the optimal solution A^*. The optimal solution involves 46 partner/lender assignments.

left vertices.

We used a sample of $194,876$ loans from the Kiva microfinance dataset [1], and created a bipartite graph. Every loan is handled by a partner (Figure 2-a). There are a total of 232 partners in the dataset that represent the left vertices of the bipartite graph and therefore the ground set E of the matroid. There are $286,874$ lenders in the dataset. We grouped these lenders into 51 clusters according to the their location: 50 representing each individual state in United States, and 1 representing all foreign lenders. These 51 lender clusters constitute the right vertices of the bipartite graph. There is an edge between a partner and a lender if the lender is among the top 50% supporters of the partner, resulting in approximately 5k edges in the bipartite graph. The weight $\bar{\mathbf{w}}(e)$ is the probability that a loan handled by partner e will be paid back. We estimate it from the dataset as $\bar{\mathbf{w}}(e) = \frac{1}{n_l}\sum_{i=1}^{n_l} \mathbf{w}_i(e)$, where n_l is the number of loans handled by this partner. We assume $\mathbf{w}_i(e)$ is 0.7 if the loan i is in repayment, 1 if it is paid, and 0 otherwise. At the beginning of each episode, we choose the loan i at random.

The optimal solution A^* is a transversal in the graph that maximizes the overall success rate of the selected partners. Top twelve partners selected based on their mean success rate in the optimal solution are shown in Figure 2-c. For each partner, the id of the lender to which this partner was assigned along with the number of eligible partners of the lender and their average success rate are listed in the Table. The objective of OMM and ε-greedy policies is similar to the optimal policy with the difference that success rates (i.e. $\mathbf{w}(e)$) are not known beforehand, and they must be learned by interacting repeatedly with the environment. Comparison results of the three policies are reported in Figure 2-b. Similar to the previous experiment, we observe the following trends. First, the expected return of OMM approaches that of the optimal solution A^* as the number of episodes increases. Second, OMM outperforms the ε-greedy policy.

5.3 Linear Matroid

In the last experiment, we evaluate OMM on the problem of learning a set of diverse and popular movies. This kind of movies is typically recommended by existing content recommender systems. The movies are popular, and therefore the user is likely to choose them. The movies are diverse, and therefore cover many potential interests of the user.

Our problem can be formulated as a *linear matroid bandit*. The ground set E are movies from the *MovieLens* dataset [16], a dataset of 6 thousand people who rated one million movies. We restrict our attention to 25 most rated movies and 75 movies that are not well known. So the cardinality of E is 100. For each movie e, we define a feature vector $\mathbf{u}_e \in \{0,1\}^{18}$, where $\mathbf{u}_e(j)$ indicates that movie e belongs to genre j. A set of movies X is considered *independent* if for any movie $e \in X$, the vector $\mathbf{u}_e$ cannot be written as a linear combination of the feature vectors of the remaining movies in X. This is our notion of diversity. The expected weight $\bar{\mathbf{w}}(e)$ is the probability that movie e is chosen. We estimate it as $\bar{\mathbf{w}}(e) = \frac{1}{n_p}\sum_{i=1}^{n_p} \mathbf{w}_i(e)$, where $\mathbf{w}_i(e)$ is the indicator that person i rated movie e and n_p is the number of people in our dataset. At the beginning of each episode, we choose the person i at random.

Twelve most popular movies from the optimal solution A^* are listed in Figure 3. These movies cover a wide range of movie genres and appear to be diverse. This validates our assumption that linear independence is suitable for modeling diversity. The expected return of OMM is reported in the same figure. We observe the same trends as in the previous experiments. More specifically, the expected return of OMM

Movie title	$\bar{\mathbf{w}}(e)$	Movie genres
American Beauty	0.568	Comedy Drama
Jurassic Park	0.442	Action Adventure Sci-Fi
Saving Private Ryan	0.439	Action Drama War
Matrix	0.429	Action Sci-Fi Thriller
Back to the Future	0.428	Comedy Sci-Fi
Silence of the Lambs	0.427	Drama Thriller
Men in Black	0.420	Action Adventure Comedy Sci-Fi
Fargo	0.416	Crime Drama Thriller
Shakespeare in Love	0.392	Comedy Romance
L.A. Confidential	0.379	Crime Film-Noir Mystery Thriller
E.T. the Extra-Terrestrial	0.376	Children's Drama Fantasy Sci-Fi
Ghostbusters	0.361	Comedy Horror

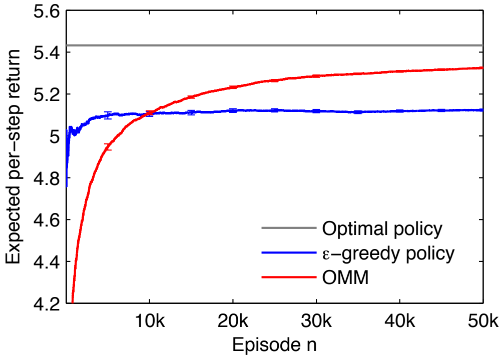


Figure 3: **Left**. Twelve most popular movies in the optimal solution A^*. The optimal solution involves 17 movies. **Right**. The expected per-step return of three movie recommendation policies in up to 50k episodes.

approaches that of A^* as the number of episodes increases and `OMM` outperforms the ε-greedy policy in 10k episodes.

6 Related Work

Our problem can be viewed as a stochastic combinatorial semi-bandit [12], where all feasible solutions are the independent sets of a matroid. Stochastic combinatorial semi-bandits were pioneered by Gai *et al.* [12], who proposed a UCB algorithm for solving these problems. Chen *et al.* [6] proved that the expected cumulative regret of this method is $O(K^2 L (1/\Delta) \log n)$. Our gap-dependent regret bound is $O(L(1/\Delta) \log n)$, a factor of K^2 tighter than the bound of Chen *et al.* [6]. Our analysis relies heavily on the properties of our problem and therefore we can derive a much tighter bound.

COMBAND [5], OSMD [2], and FPL [19] are algorithms for adversarial combinatorial semi-bandits. The main limitation of COMBAND and OSMD is that they are not guaranteed to be computationally efficient. More specifically, COMBAND needs to sample from a distribution over exponentially many solutions and OSMD needs to project to the convex hull of these solutions. FPL is computationally efficient but not very practical because its time complexity increases with time. On the other hand, `OMM` is guaranteed to be computationally efficient but can only solve a special class of combinatorial bandits, matroid bandits.

Matroids are a broad and important class of combinatorial optimization problems [21], which has been an active area of research for the past 80 years. This is the first paper that studies a well-known matroid problem in the bandit setting and proposes a learning algorithm for solving it.

Our work is also related to submodularity [18]. In particular, let:

$$g(X) = \max_{Y : Y \subseteq X, Y \in \mathcal{I}} f(Y, \bar{\mathbf{w}}) \tag{29}$$

be the maximum weight of an independent set in X. Then it is easy to show that $g(X)$ is submodular and monotonic in X, and that the maximum-weight basis of a matroid is a solution to $A^* = \arg\max_{A : |A| = K} g(A)$. Many algorithms for learning how to maximize a submodular function have been proposed recently [13, 26, 10, 24, 11]. None of these algorithms are suitable for solving our problem. There are two reasons. First, each algorithm is designed to maximize a specific submodular function and our function g may not be of that type. Second, the algorithms are only near optimal, learn a set A such that $g(A) \geq (1 - 1/e) g(A^*)$. Note that our method is guaranteed to be optimal and learn A^*.

7 Conclusions

This is the first work that studies the problem of learning a maximum-weight basis of a matroid, where the weights of the items are initially unknown, and have to be learned by interacting repeatedly with the environment. We propose a practical algorithm for solving this problem and bound its regret. The regret is sublinear in time and at most linear in all other quantities of interest. We evaluate our method on three real-world problems and show that it is practical.

Our regret bounds are $\Omega(\sqrt{L})$. Therefore, `OMM` is not practical when the number of items L is large. We believe that these kinds of problems can be solved efficiently by introducing additional structure, such as *linear generalization*. In this case, the weight of each item would be modeled as a linear function of its features and the goal is to learn the parameters of this function.

Many combinatorial optimization problems can be viewed as optimization on a matroid or its generalizations, such as *maximum-weight matching* on a bipartite graph and *minimum cost flows*. In a sense, these are the hardest problems in combinatorial optimization that can be solved optimally in polynomial time [22]. In this work, we show that one of these problems is efficiently learnable. We believe that the key ideas in our work are quite general and can be applied to other problems that involve matroids.

References

[1] KIVA. http://build.kiva.org/docs/data, 2013.

[2] Jean-Yves Audibert, Sebastien Bubeck, and Gabor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[4] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012.

[5] Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[6] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

[7] Baek-Young Choi, Sue Moon, Zhi-Li Zhang, Konstantina Papagiannaki, and Christophe Diot. Analysis of point-to-point packet delay in an operational network. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.

[8] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[9] Jack Edmonds and Delbert Ray Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards*, 69B(3):147–153, 1965.

[10] Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems 26*, pages 2697–2705, 2013.

[11] Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Large-scale optimistic adaptive submodularity. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

[12] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

[13] Andrew Guillory and Jeff Bilmes. Online submodular set cover, ranking, and repeated active learning. In *Advances in Neural Information Processing Systems 24*, pages 1107–1115, 2011.

[14] Naoki Katoh. Combinatorial optimization algorithms in resource allocation problems. *Encyclopedia of Optimization*, pages 259–264, 2001.

[15] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[16] Shyong Lam and Jon Herlocker. MovieLens 1M Dataset. http://www.grouplens.org/node/12, 2012.

[17] Rémi Munos. The optimistic principle applied to games, optimization, and planning: Towards foundations of Monte-Carlo tree search. *Foundations and Trends in Machine Learning*, 2012.

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1):265–294, 1978.

[19] Gergely Neu and Gábor Bartók. An efficient algorithm for learning with semi-bandit feedback. In *Proceedings of the 24th International Conference on Algorithmic Learning Theory*, pages 234–248, 2013.

[20] Carlos Oliveira and Panos Pardalos. A survey of combinatorial optimization problems in multicast routing. *Computers and Operations Research*, 32(8):1953–1981, 2005.

[21] James Oxley. *Matroid Theory*. Oxford University Press, New York, NY, 2011.

[22] Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization*. Dover Publications, Mineola, NY, 1998.

[23] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with Rocketfuel. *IEEE / ACM Transactions on Networking*, 12(1):2–16, 2004.

[24] Zheng Wen, Branislav Kveton, Brian Eriksson, and Sandilya Bhamidipati. Sequential Bayesian search. In *Proceedings of the 30th International Conference on Machine Learning*, pages 977–983, 2013.

[25] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

[26] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems 24*, pages 2483–2491, 2011.