Approximation Algorithms for Online Weighted Rank Function Maximization under Matroid Constraints

Niv Buchbinder[1], Joseph (Seffi) Naor[2], R. Ravi[3], and Mohit Singh[4]

[1] Computer Science Dept., Open University of Israel. E-mail: niv.buchbinder@gmail.com
[2] Computer Science Dept., Technion, Haifa, Israel. E-mail: naor@cs.technion.ac.il
[3] Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA. E-mail: ravi@cmu.edu. Supported in part by NSF award CCF-1143998
[4] Microsoft Research, Redmond and School of Computer Science, McGill University, Montreal, Quebec, Canada. E-mail: mohitsinghr@gmail.com

Abstract. Consider the following online version of the submodular maximization problem under a matroid constraint: We are given a set of elements over which a matroid is defined. The goal is to incrementally choose a subset that remains independent in the matroid over time. At each time, a new weighted rank function of a different matroid (one per time) over the same elements is presented; the algorithm can add a few elements to the incrementally constructed set, and reaps a reward equal to the value of the new weighted rank function on the current set. The goal of the algorithm as it builds this independent set online is to maximize the sum of these (weighted rank) rewards. As in regular online analysis, we compare the rewards of our online algorithm to that of an offline optimum, namely a single independent set of the matroid that maximizes the sum of the weighted rank rewards that arrive over time. This problem is a natural extension of two well-studied streams of earlier work: the first is on online set cover algorithms (in particular for the max coverage version) while the second is on approximately maximizing submodular functions under a matroid constraint.

In this paper, we present the first randomized online algorithms for this problem with poly-logarithmic competitive ratio. To do this, we employ the LP formulation of a scaled reward version of the problem. Then we extend a weighted-majority type update rule along with uncrossing properties of tight sets in the matroid polytope to find an approximately optimal fractional LP solution. We use the fractional solution values as probabilities for a online randomized rounding algorithm. To show that our rounding produces a sufficiently large reward independent set, we prove and use new covering properties for randomly rounded fractional solutions in the matroid polytope that may be of independent interest.

1 Introduction

Making decisions in the face of uncertainty is the fundamental problem addressed by online computation [5]. In many planning scenarios, a planner must decide

on the evolution of features to a product without knowing the evolution of the demand for these features from future users. Moreover, any features initially included must be retained for backward compatibility, and hence leads to an online optimization problem: given a set of features, the planner must phase the addition of the features, so as to maximize the value perceived by a user at the time of arrival. Typically, users have diminishing returns for additional features, so it is natural to represent their utility as a submodular function of the features that are present (or added) when they arrive. Furthermore, the set of features that are thus monotonically added, are typically required to obey some design constraints. The simplest are of the form that partition the features into classes and there is a restriction on the number of features that can be deployed in each class. A slight extension specifies a hierarchy over these classes and there are individual bounds over the number of features that can be chosen from each class. We capture these, as well as other much more general restrictions on the set of deployed features, via the constraint that the chosen features form an independent set of a matroid. Thus, our problem is to monotonically construct an independent set of features (from a matroid over the features) online, so as to maximize the sum of submodular function values (users) arriving over time and evaluated on the set of features that have been constructed so far.

This class of online optimization problems generalizes some early work of Awerbuch et al. [2]. They considered a set-cover instance, in which the restriction is to choose at most k sets with the goal of maximizing the coverage of the elements as they arrive over time. This is precisely the online maximization version of the well-studied maximum coverage problem. Even this special case of our problem already abstracts problems in investment planning, strategic planning, and video-on-demand scheduling.

1.1   Problem Setting, Main Result and Techniques

In our setting[5], we are given a universe of elements E, $|E| = m$, and a matroid $\mathcal{M} = (E, \mathcal{I}(\mathcal{M}))$ whose independent sets characterize the limitations on which sets of elements we can choose. At every time step i, $1 \le i \le n$, a client arrives with a non-negative monotone submodular function $f_i : 2^E \to \mathbb{Z}_+$ representing her welfare function. The objective is to maintain a monotonically increasing set $F \in \mathcal{I}(\mathcal{M})$ over time; that is, the set F_{i-1} of elements (at time $i-1$) can only be augmented to F_i after seeing f_i at time step i. The welfare of client i is then $f_i(F_i)$, and our objective is to maximize $\sum_{i=1}^{n} f_i(F_i)$. We compare our performance to the offline optimum $\max_{O \in \mathcal{I}(\mathcal{M})} \sum_{i=1}^{n} f_i(O)$.

In this paper, we are concerned with the case when each of the submodular functions f_i is a weighted rank function of some matroid $\mathcal{N}_i$, i.e., $f_i(S) = \max_{I \subseteq S, I \in \mathcal{I}(\mathcal{N}_i)} \sum_{e \in I} w_{i,e}$ where $w_i : E \to \mathbb{R}_+$ is an arbitrary weight function. This class of submodular functions is very broad and includes all the examples discussed above; Furthermore, we believe it captures the difficulty of general

---

[5] For preliminaries and basic definitions, please see Section 2.

submodular functions even though we have not yet been able to extend our results to the general case. Nevertheless, there are submodular functions which are not weighted rank functions of a matroid, for example, multi-set coverage function [7].

Theorem 1. *There exists a randomized polynomial time algorithm which is $O\left(\log^2 n \log m \log f_{ratio}\right)$-competitive, for the online submodular function maximization problem under a matroid constraint over m elements, when each f_i, $1 \leq i \leq n$, is a weighted rank function of a matroid and $f_{ratio} = 2\frac{\max_{i,e} f_i(\{e\})}{\min_{i,e|f_i(\{e\})\neq 0} f_i(\{e\})}$. In other words, the algorithm maintains monotonically increasing independent sets $F_i \in \mathcal{I}(\mathcal{M})$ such that*

$$E\left[\sum_{i=1}^{n} f_i(F_i)\right] \geq \Omega\left(\frac{1}{\log^2 n \log m \log f_{ratio}}\right) \cdot \max_{O \in \mathcal{I}(\mathcal{M})} \sum_{i=1}^{n} f_i(O).$$

Our result should be contrasted with the lower bound proved in [2][6].

Lemma 1. *[2] Any randomized algorithm for the submodular maximization problem under a matroid constraint is $\Omega(\log n \log(m/r))$-competitive, where r is the rank of the matroid. This lower bound holds even for uniform matroids and when all f_i are unweighted rank functions.*

We note that the $O(\log m)$ factor in our analysis can be improved slightly to an $O(\log(m/r))$ factor with a more careful analysis. A lower bound of $\Omega(\log f_{ratio})$ also follows even when the functions f_i are linear (see, for example, [6]).

Main Techniques. To prove our results, we combine techniques from online computation and combinatorial optimization. The first step is to formulate an integer linear programming formulation for the problem. Unfortunately, the natural linear program is not well-suited for the online version of the problem. Thus, we formulate a different linear program in which we add an extra constraint that each element e *contributes* roughly the same value to the objective of the optimal solution. While this may not be true in general, we show that an approximate optimal solution satisfies this requirement.

We note that the online setting we study is quite different from the online packing framework studied by [6] and leads to new technical challenges. In particular, there are two obstacles in applying the primal-dual techniques in [6] to our setting. First, the linear formulation we obtain (that is natural for our problem) is not a strict packing LP and contains negative variables (See Section 3). Second, the number of packing constraints is exponential, and hence the techniques of [6] would give a linear competitive factor rather than a polylogarithmic one. Nevertheless, we present in Section 3 an online algorithm which gives a fractional solution to the linear program having a large objective value. One of the crucial ingredients is the *uncrossing* property of tight sets for any feasible point in the matroid polytope.

---

[6] The lower bounds in [2] relate even to a special case of uniform matroid and very restricted sub-modular functions.

To obtain an integral solution, we perform in Section 4 a natural randomized rounding procedure to select fractionally chosen elements. But, we have to be careful to maintain that the selected elements continue to form an independent set. The main challenge in the analysis is to tie the performance of the randomized algorithm to the performance of the fractional algorithm. As a technical tool in our proof, we show in Lemma 9 that randomly rounding a fractional solution in the matroid polytope gives a set which can be covered by $O(\log n)$ independent sets with high probability. This lemma may be of independent interest and similar in flavor to the results of Karger [14] who proved a similar result for packing bases in the randomly rounded solution.

1.2 Related results

Maximizing monotone submodular function under matroid constraints has been a well studied problem and even many special cases have been studied widely (see survey by Goundan and Schulz [13]). Fisher, Nemhauser and Wolsey [12] gave a $(1 - \frac{1}{e})$-approximation when the matroid is the uniform matroid and showed that the greedy algorithm gives a $\frac{1}{2}$-approximation. This was improved by Calinescu *at al* [7] and Vondrák [20] who gave a $(1 - \frac{1}{e})$-approximation for the general problem. They also introduced the *multi-linear extension* of a submodular function and used pipage rounding introduced by Ageev and Sviridenko [1]. The facility location problem was introduced by Cornuejols et al. [9] and was the impetus behind studying the general submodular function maximization problem subject to matroid constraints. The submodular welfare problem can be cast as a submodular maximization problem subject to a matroid constraint and the reduction appears in Fisher *et al.* [12] and the problem has been extensively studied [18, 16, 17, 15]. The result of Vondrák [20] implies a $(1 - \frac{1}{e})$-approximation for the problem. Despite the restricted setting of our benefit functions, we note that recent work in welfare maximization in combinatorial auctions [10] has focused on precisely the case when the valuations are matroid rank sums (MRS) that we consider in our model.

A special case of our online problem was studied by Awerbuch et al. [2]. They studied an online variant of the max-coverage problem, where given n sets covering m elements, the elements arrive one at a time, and the goal is to pick up to k sets online to maximize coverage. They obtained a randomized algorithm that is $O(\log n \log(m/k))$-competitive for the problem and proved that this is optimal in their setting. Our results generalize both the requirement on the cardinality of the chosen sets to arbitrary matroid constraints, and the coverage functions of the arriving elements to monotone submodular functions that are weighted rank functions of matroids.

Another closely related problem with a different model of uncertainty was studied by Babaioff et al. [3]. They studied a setting in which elements of a matroid arrive in an online fashion and the goal is to construct an independent set that is competitive with the maximum weight independent set. They considered the random permutation model which is a non-adversarial setting, and obtained an $O(\log k)$-competitive algorithm for general matroid, where k is the rank of

the matroid, and constant competitive ratio for several interesting matroids. Recently, Bateni et al. [4] studied the same model where the objective function is a submodular function (rather than linear).

Chawla et al. [8] study Bayesian optimal mechanism design to maximize expected revenue for a seller while allocating items to agents who draw their values for the items from a known distribution. Their development of agent-specific posted price mechanisms when the agents arrive in order, and the items allocated must obey matroid feasibility constraints, is similar to our setting. In particular, we use the ideas about certain ordering of matroid elements (Lemma 7 in their paper) in the proof that our randomized rounding algorithm give sufficient profit.

2 Preliminaries

Given a set E, a function $f : 2^E \to \mathbb{R}_+$ is called *submodular* if for all sets $A, B \subseteq E$, $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$. Given set E and a collection $\mathcal{I} \subseteq 2^E$, $\mathcal{M} = (E, \mathcal{I}(\mathcal{M}))$ is a *matroid* if (i) for all $A \in \mathcal{I}$ and $B \subset A$ implies that $B \in \mathcal{I}$ and (ii) for all $A, B \in \mathcal{I}$ and $|A| > |B|$ then there exists $a \in A \setminus B$ such that $B \cup \{a\} \in \mathcal{I}$. Sets in $\mathcal{I}$ are called *independent sets* of the matroid $\mathcal{M}$. The rank function $r : 2^E \to R^+$ of matroid $\mathcal{M}$ is defined as $r(S) = \max_{T \in \mathcal{I} : T \subseteq S} |T|$. A standard property of matroids is the fact that the rank function of any matroid is submodular.

We also work with weighted rank functions of a matroid, defined as $f(S) = \max_{I \subseteq S, I \in \mathcal{I}(\mathcal{M})} \sum_{e \in I} w_e$ for some weight function $w : 2^E \to \mathbb{R}_+$. Given any matroid $\mathcal{M}$, we define the matroid polytope to be the convex hull of independent sets $P(\mathcal{M}) = conv\{\mathbf{1_I} : \mathbf{I} \in \mathcal{I}\} \subseteq \mathbb{R}^{|\mathbf{E}|}$. Edmonds [11] showed that $P(\mathcal{M}) = \{x \geq 0 : x(S) \leq r(S) \ \forall \, S \subseteq E\}$. We also use the following fact about fractional points in the matroid polytope (The proof follows from standard uncrossing arguments. See Schrijver [19], Chapter 40).

Fact 2 *Given a matroid $\mathcal{M} = (E, \mathcal{I}(\mathcal{M}))$ with rank function r and feasible point $x \in P(\mathcal{M})$, let $\tau = \{S \subseteq E : x(S) = r(S)\}$. Then, τ is closed under intersection and union and there is a single maximal set in τ.*

3 Linear Program and the Fractional Algorithm

We now give a linear program for the online submodular function maximization problem and show how to construct a feasible fractional solution online which is $O(\log m \log n \log f_{ratio})$-competitive. Before we give the main theorem, we first formulate a natural LP. Let $O \subseteq E$ denote the optimal solution with the objective $\sum_{i=1}^n f_i(O)$. Since each f_i is the weighted rank function of matroid $\mathcal{N}_i$, we have that $f_i(O) = w_i(O_i) = \sum_{e \in O_i} w_{i,e}$ where $O \supseteq O_i \in \mathcal{I}(\mathcal{N}_i)$. For the sake of simplicity, we assume that $w_{i,e} = 1$ (In the full version we show that this assumption can be removed with a loss of $O(\log f_{ratio})$ factor in the competitive

$$LP_1: \qquad \max \ \sum_{i=1}^{n} \sum_{e \in E} z_{i,e}$$
$$\text{s.t.}$$

$$\forall S \subseteq E \quad \sum_{e \in S} x_e \leq r(S) \qquad\qquad (1)$$

$$\forall 1 \leq i \leq n, S \subseteq E \ \sum_{e \in S} z_{i,e} \leq r_i(S) \qquad\qquad (2)$$

$$\forall 1 \leq i \leq n, e \in E \qquad z_{i,e} \leq x_e \qquad\qquad (3)$$

$$\forall 1 \leq i \leq n, e \in E \qquad z_{i,e}, x_e \geq 0$$

Fig. 1. LP for maximizing a sum of (unweighted) rank functions subject to matroid constraint

$$LP_2(\alpha): \qquad \max \ \sum_{i=1}^{n} \sum_{e \in E} z_{i,e}$$
$$\text{s.t.}$$

$$\forall S \subseteq E \quad \sum_{e \in S} x_e \leq r(S) \qquad\qquad (4)$$

$$\forall 1 \leq i \leq n, S \subseteq E \ \sum_{e \in S} z_{i,e} \leq r_i(S) \qquad\qquad (5)$$

$$\forall 1 \leq i \leq n, e \in E \qquad z_{i,e} \leq x_e \qquad\qquad (6)$$

$$\forall e \in E \quad \sum_{i=1}^{n} z_{i,e} \leq \alpha x_e \qquad\qquad (7)$$

$$\forall e \in E \quad \sum_{i=1}^{n} z_{i,e} \geq \frac{\alpha x_e}{2} \qquad\qquad (8)$$

$$\forall 1 \leq i \leq n, e \in E \qquad z_{i,e}, x_e \geq 0$$

Fig. 2. A restricted LP for the submodular function maximization subject to matroid constraint

ratio). Observe that in this case, $f_i(S) = r_i(S)$, where r_i is the rank function of matroid $\mathcal{N}_i$ for any set $S \subseteq E$.

We next formulate a linear program where x_e is the indicator variable for whether $e \in O$ and $z_{i,e}$ is the indicator variable for whether $e \in O_i$. Since $O \in \mathcal{I}(M)$ and $O_i \in \mathcal{I}(\mathcal{N}_i)$, we have that $x \in P(\mathcal{M})$ and $z_i \in P(\mathcal{N}_i)$ as represented by constraints (1) and constraints (2), respectively in Figure 1.

We prove the following theorem.

Theorem 3. *There exists a polynomial time algorithm $\mathcal{A}$ that constructs a feasible fractional solution (x, z) online to LP_1 which is $O(\log n \log m)$-competitive. That is, the algorithm $\mathcal{A}$ maintains monotonically increasing solution (x, z) such that $\sum_{i=1}^{n} \sum_{e \in E} z_{i,e} = \Omega(\frac{\sum_{i=1}^{n} f_i(O)}{\log n \log m})$ where O is the optimal integral solution.*

To prove Theorem 3, instead of working with the natural linear program LP_1, we formulate a different linear program. The new linear program is indexed by an integer α and places the constraints that each $e \in O$ occurs in $[\frac{\alpha}{2}, \alpha]$ different O_i's as represented by constraints (7) and (8). The parameter α will be defined later.

The following lemma whose proof is omitted shows that if we pick $O(\log n)$ different values of α then the sum of the integer solutions to the linear programs $LP_2(\alpha)$ perform as well as the optimal solution.[7]

Lemma 2. *Let OPT denote the value of an optimal integral solution to linear program LP_1 and let OPT_α denote the optimal value of an optimal integral solution to the linear program $LP_2(\alpha)$ for each $\alpha \in \{1, 2, 4, \ldots, 2^{\lceil \log n \rceil}\}$. Then $OPT \leq \sum_{\alpha \in \{1, 2, 4, \ldots, 2^{\lceil \log n \rceil}\}} OPT_\alpha$.*

Using the above lemma, a simple averaging argument shows that for some guess α, the optimal integral solution to $LP_2(\alpha)$ is within a $\log n$ factor of the optimal integral solution to LP_1. Hence, we construct an algorithm which first guesses α and then constructs an approximate fractional solution to $LP_2(\alpha)$.

3.1 Online Algorithm for a Fractional LP Solution

Given a fractional solution x, we call a set $S \subseteq E$ tight (with respect to x) if $x(S) = r(S)$.

Guessing Algorithm:

– Guess the value $\alpha \in_R \{1, 2, 4 \ldots, n\}$.
– Run AlgG with value α.

AlgG:

– Initialize $x_e \leftarrow 1/m^2$ (where $m = |E|$), set $z_{i,e} = 0$ for each i, e.
– When function f_i arrives, order the elements arbitrarily.
– For each element e in order:
– If $\forall S | e \in S,\ x(S) < r(S)$ and $z_i(S) < r_i(S) - 1/2$:

$$x_e \leftarrow \min \left\{ x_e \cdot \exp\left(\frac{8 \log m}{\alpha}\right), \min_{S | e \in S} \{r(S) - x(S \setminus \{e\})\} \right\} \quad (9)$$

$$z_{i,e} \leftarrow x_e/2 \quad (10)$$

Using an independence oracle for each of the matroids, each of the conditions can be checked in polynomial time by reducing it to submodular function minimization (See Schrijver [19], Chapter 40) and therefore the running time of the algorithm is polynomial. Note that the fractional algorithm is carefully designed. For example, it is very reasonable to update greedily the value of $z_{i,e}$ even when the algorithm does not update the value x_e (of course, ensuring that $z_i \in P(\mathcal{N}_i)$). While such an algorithm does give the required guarantee on the performance of the fractional solution, it is not clear how to round such a solution to an integral solution. In particular, our algorithm for finding a fractional

---

[7] We assume that the algorithm knows the value of n. In the full version of the paper we show how to deal with an unknown n losing an additional small factor.

solution is tailored so as to allow us to use the values as rounding probabilities in a randomized algorithm.

Before we continue, we define some helpful notation regarding the online algorithm. Let $x_{i,e}(\alpha)$ be the value of the variable x_e after the arrival of user i for some guess α. Let $\Delta x_{i,e}(\alpha)$ be the change in the value of x_e when user i arrives. Let $x_e(\alpha)$ be the value of x_e at the end of the execution. Similarly, let $z_{i,e}(\alpha)$ be the value of $z_{i,e}$ at the end of the execution. We start with the following lemma that follows from the update rule (9). The proof is omitted.

Lemma 3. *For any element $e \in E$, and guess α,*

$$\sum_{i=1}^{n} z_{i,e}(\alpha) \geq \frac{\alpha}{48 \log m} \left(x_e(\alpha) - \frac{1}{m^2} \right) \tag{11}$$

where $x_e(\alpha)$ is the value at the end of the execution of AlgG.

Next we prove that the solution produced by AlgG is almost feasible with respect to the optimal solution to $LP_2(\alpha)$.

Lemma 4 (Feasibility Lemma). *Let $(x(\alpha), z(\alpha))$ be the fractional solution generated by AlgG at the end of the sequence. Then, it satisfies all constraints of $LP_2(\alpha)$ except constraints (8).*

Proof. We prove that the solution is feasible.
Matroid constraints (4). Clearly, the algorithm never violates the matroid constraints by the second term in the equation (9) in the algorithm.
Constraints (5) and constraints (6). $z_{i,e} \leftarrow x_{i,e}(\alpha)/2 \leq x_e(\alpha)/2$, thus constraints (6) hold. Finally, by the algorithm behavior we only update $z_{i,e}$ if for all $S|e \in S$, $z_i(S) < r_i(S) - 1/2$. Since by the above observations $z_{i,e} \leq x_e(\alpha)/2 \leq 1/2$, we never violate constraints (5) after the update.
Constraints (7). This constraint follows since

$$\sum_{i=1}^{n} z_{i,e} = \sum_{i:\Delta x_{i,e}>0} x_{i,e}(\alpha)/2 \leq x_e(\alpha)|\{i : \Delta x_{i,e} > 0\}|$$

However, after α augmentations, $x_e(\alpha) \geq \frac{1}{m^2} \exp\left(\frac{8 \log m}{\alpha} \cdot \alpha \right) > 1$. Thus, x_e must be in a tight set and so by design we never update x_e and any $z_{i,e}$.

In order to evaluate the performance of the algorithm we first show that the *size* of the solution returned by the algorithm is large as compared to the optimal integral solution. Later in Lemma 6, we relate the objective value of the solution to its size. This lemma uses crucially the properties of the matroid. The proof is omitted.

Lemma 5 (Large Fractional Size). *Let $(x^*(\alpha), z^*(\alpha))$ be an optimal integral solution to $LP_2(\alpha)$. Let $(x(\alpha), z(\alpha))$ be the fractional solution generated by AlgG at the end of the sequence. Then we have, $\sum_{e \in E} x_e(\alpha) \geq \frac{1}{16} \sum_{e \in E} x_e^*(\alpha)$.*

Finally, we prove a lemma bounding the performance of the algorithm.

Lemma 6. *For any guess value α, the algorithm maintains a fractional solution to $LP_2(\alpha)$ such that:*

$$\sum_{e \in E} \sum_{i=1}^{n} z_{i,e}(\alpha) = \Omega\left(\frac{OPT_\alpha}{\log m}\right),$$

where OPT_α is objective of an optimal integral solution to $LP_2(\alpha)$.

Proof. Let (x^*, z^*) denote the optimal integral solution to $LP_2(\alpha)$. If $x_e^* = 0$ for each e, then the lemma follows immediately. We have the following

$$\sum_{e \in E} \sum_{i=1}^{n} z_{i,e}(\alpha) \geq \quad \frac{\alpha}{48 \log m} \sum_{e \in E} \left(x_e(\alpha) - \frac{1}{m^2}\right) \quad (Lemma\ 3)$$

$$\geq \quad \frac{\alpha}{48 \log m} \sum_{e \in E} \left(\frac{x_e^*(\alpha)}{16} - \frac{1}{m^2}\right) \quad (Lemma\ 5)$$

$$= \Omega\left(\frac{1}{\log m} \sum_{e \in E} \sum_{i=1}^{n} z_{i,e}^*(\alpha)\right)$$

where the last equality follows since in $LP_2(\alpha)$ for each element $\sum_{i=1}^{n} z_{i,e}^*(\alpha) \leq \alpha x_e^*$ and $\sum_{e \in E} x_e^* \geq 1$. This completes the proof of Lemma.

Finally, we get our main theorem.

Theorem 4. *The online algorithm for the fractional LP solution (of LP_1) is $O(\log m \log n)$-competitive.*

Proof. The proof follows by combining Lemma (4), Lemma (6), Lemma (2) and the observation that there are $O(\log n)$ possible values of α each is guessed with probability $\Omega(1/\log n)$.

4 Randomized Rounding Algorithm

In this section we present a randomized algorithm for the unweighted problem that is $O(\log^2 n \log m)$-competitive when each submodular function f_i is a rank function of a matroid. The algorithm is based on the fractional solution designed in Section 3. Although our rounding scheme is extremely simple, the proof of its correctness involves carefully matching the performance of the rounding algorithm with the performance of the fractional algorithm. Indeed, here the fact that $LP_2(\alpha)$ has extra constraints not present in LP_1 is used very crucially.

Theorem 5. *The expected profit of the randomized algorithm is $\Omega\left(\frac{OPT}{\log m \log^2 n}\right)$.*

The randomized algorithm follows the following simple rounding procedure.

Matroid Randomized Rounding Algorithm:

- $F \leftarrow \emptyset$.
- Guess the value $\alpha \in_R \{1, 2, 4 \dots, n\}$.
- Run AlgG with value α.
  - Whenever x_e increases by Δx_e, if $F \cup \{e\} \in \mathcal{I}(\mathcal{M})$ then $F \leftarrow F \cup \{e\}$ with probability $\frac{\Delta x_e}{4}$.

In order to prove our main theorem, we prove several crucial lemmas. The main idea is to tie the performance of the randomized algorithm to the performance of the fractional solution that is generated. In the process we lose a factor of $O(\log n)$. We first introduce some notation. All of the following notation is with respect to the execution of the online algorithm for a fixed value of α and we omit it from the notation. Let F_i denote the solution formed by the randomized algorithm at the end of iteration i and let F denote the final solution returned by the algorithm. Let Y_e^i denote the indicator random variable that element e has been selected *till* iteration i. Let ΔY_e^i denote the indicator random variable that element e is selected in iteration i. Let $y_e^i = Pr[Y_e^i = 1]$ and $\Delta y_e^i = Pr[\Delta Y_e^i = 1]$. Finally, let y_e denote the probability element e is in the solution at the end of the execution. Recall that $x_{i,e}$ denotes the value of the variable x_e in the fractional solution after iteration i and let x_e denote the fractional value of element e at the end of the execution of the fractional algorithm, and let $\Delta x_{i,e}$ be the change in the value of e in iteration i.

Since the algorithm tosses a coin for element e in iteration i with probability $\Delta x_{i,e}/4$, therefore the probability that an element e is included in the solution till iteration i is at most $x_{i,e}/4$. Our first lemma states that the expected number of elements chosen by the algorithm is at least half that amount in expectation and is comparable to the total size of the fractional solution. Thus, Lemma 7 plays the role of Lemma 5 in the analysis of the randomized algorithm. The proof is omitted.

Lemma 7. *Let F be the solution returned by the randomized rounding algorithm, then $E[|F|] = \sum_{e \in E} y_e \geq \frac{\sum_{e \in E} x_e}{8}$.*

Our second lemma relates the change in the probability an element is chosen to the change in the fractional solution. This lemma shows that a crucial property of the exponential update rule for the fractional solution is also satisfied by the integral solution. The proof is omitted.

Lemma 8. *For each element e and iteration i, $\frac{\Delta y_e^i}{y_e^i} \leq \frac{\Delta x_{i,e}}{x_{i,e}} \leq \frac{24 \log m}{\alpha}$.*

We next prove a general lemma regarding randomized rounding in any matroid polytope. The proof of the lemma that utilizes a lemma proven in Chawla et al. [8] is omitted.

Lemma 9. *Given a matroid $\mathcal{N} = (E, \mathcal{I})$ and a solution z such that for all $S \subseteq E$, $z(S) \leq r(S)/2$, construct a set F by including in $e \in F$ with probability z_e for each $e \in E$ independently. Then, with high probability $(1 - \frac{1}{m^2 n^2})$, F can be covered by $O(\log m + \log n)$ independent sets where $m = |N|$.*

We now prove a relation between the profit obtained by the algorithm at iteration i, denoted by the random variable $r_i(F_i)$, and the events that a particular set of elements are chosen in the solution. For any i, Let H_i denote the set of elements such that $z_{i,e} > 0$. Note that $z_{i,e} > 0$ iff $\Delta x_{i,e} > 0$.

Lemma 10. $\sum_{i=1}^{n} E[r_i(F_i)] \geq \frac{1}{c \log n} \sum_{i=1}^{n} \sum_{e \in H_i} y_e^i$, where c is some constant.

Now we have all the ingredients to prove Theorem 5.

Proof of Theorem 5: We prove that the expected profit of the algorithm with a guess α is at least $\Omega\left(\frac{OPT_\alpha}{\log m \log n}\right)$. Since each α is guessed with probability $1/\log n$ and the value of OPT is the sum over all values α we get the desired. The expected profit of the algorithm when we guess α is at least.

$$
\begin{aligned}
\sum_{i=1}^{n} E[f_i(F_i)] &\geq \tfrac{1}{c \log n} \sum_{i=1}^{n} \sum_{e \in H_i} y_e^i && (Lemma\ 10) \\
&\geq \sum_{i=1}^{n} \sum_{e \in H_i} \tfrac{\alpha}{c' \log m \log n} \Delta y_e^i && (Lemma\ 8) \\
&= \sum_{e \in E} \tfrac{\alpha}{c' \log m \log n} y_e && (\textstyle\sum_{i : e \in H_i} \Delta y_e^i = y_e) \\
&\geq \sum_{e \in E} \tfrac{\alpha}{8c' \log m \log n} x_e && (Lemma\ 7) \\
&= \Omega\left(\tfrac{\alpha \cdot n_\alpha}{\log m \log n}\right) = \Omega\left(\tfrac{OPT_\alpha}{\log m \log n}\right) && (Lemma\ 5)
\end{aligned}
$$

$\square$

References

1. Alexander A. Ageev and Maxim Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim.*, 8(3):307–328, 2004.
2. Baruch Awerbuch, Yossi Azar, Amos Fiat, and Tom Leighton. Making commitments in the face of uncertainty: how to pick a winner almost every time (extended abstract). In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 519–530, 1996.
3. Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443, 2007.
4. MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. In *APPROX-RANDOM*, pages 39–52, 2010.
5. Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 1998.
6. Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2009.
7. Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In *IPCO '07: Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*, pages 182–196, 2007.

8. Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *ACM Symposium on Theory of Computing*, pages 311–320, 2010.

9. Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

10. Shaddin Dughmi, Tim Roughgarden, and Qiqi Yan. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *ACM Symposium on Theory of Computing*, pages 149–158, 2011.

11. J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Proceedings of the Calgary International Conference on Combinatorial Structures and their Application, Gordon and Breach, New York*, pages 69–87, 1969.

12. Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. An analysis of approximations for maximizing submodular set functions - part ii. *Mathematical Programming*, 14:265–294, 1978.

13. P.R. Goundan and A.S. Schulz. Revisiting the greedy approach to submodular set function maximization. *Preprint*, January 2009.

14. David R. Karger. Random sampling and greedy sparsification for matroid optimization problems. *Mathematical Programming*, 82:99–116, 1998.

15. Subhash Khot, Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. *Algorithmica*, 52(1):3–18, 2008.

16. Benny Lehmann, Daniel J. Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM Conference on Electronic Commerce*, pages 18–28, 2001.

17. Vahab S. Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *ACM Conference on Electronic Commerce*, pages 70–77, 2008.

18. G. L. Nemhauser and L. A. Wolsey. Best Algorithms for Approximating the Maximum of a Submodular Set Function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

19. A. Schrijver. Combinatorial optimization - polyhedra and efficiency. *Springer-Verlag*, 2005.

20. Jan Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 67–74, 2008.