

Approximating Submodular Functions Everywhere

Michel X. Goemans* Nicholas J. A. Harvey†

Satoru Iwata‡ Vahab Mirrokni§

Abstract

Submodular functions are a key concept in combinatorial optimization. Algorithms that involve submodular functions usually assume that they are given by a (value) oracle. Many interesting problems involving submodular functions can be solved using only polynomially many queries to the oracle, e.g., exact minimization or approximate maximization.

In this paper, we consider the problem of approximating a non-negative, monotone, submodular function f on a ground set of size n *everywhere*, after only $\text{poly}(n)$ oracle queries. Our main result is a deterministic algorithm that makes $\text{poly}(n)$ oracle queries and derives a function \hat{f} such that, for *every* set S , $\hat{f}(S)$ approximates $f(S)$ within a factor $\alpha(n)$, where $\alpha(n) = \sqrt{n+1}$ for rank functions of matroids and $\alpha(n) = O(\sqrt{n} \log n)$ for general monotone submodular functions. Our result is based on approximately finding a maximum volume inscribed ellipsoid in a symmetrized polymatroid, and the analysis involves various properties of submodular functions and polymatroids.

Our algorithm is tight up to logarithmic factors. Indeed, we show that no algorithm can achieve a factor better than $\Omega(\sqrt{n}/\log n)$, even for rank functions of a matroid.

1 Introduction

Let $f: 2^{[n]} \rightarrow \mathbb{R}_+$ be a function where $[n] = \{1, 2, \dots, n\}$. The function f is called *submodular* if

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T),$$

for all $S, T \subseteq [n]$. Additionally, f is called *monotone* if $f(Y) \leq f(Z)$ whenever $Y \subseteq Z$. An equivalent definition of submodularity is the property of *decreasing marginal values*: For any $Y \subseteq Z \subseteq [n]$ and $x \in [n] \setminus Z$, $f(Z \cup \{x\}) - f(Z) \leq f(Y \cup \{x\}) - f(Y)$. This can be deduced from the first definition by substituting

$S = Y \cup \{x\}$ and $T = Z$; the reverse implication also holds [28, §44.1]. We assume a *value oracle* access to the submodular function; i.e., for a given set S , an algorithm can query an oracle to find its value $f(S)$.

Background. Submodular functions are a key concept in operations research and combinatorial optimization, see for example the books [10, 28, 26]; the term ‘submodular’ has over 500 occurrences in Schrijver’s 3-volume book on combinatorial optimization [28]. Submodular functions are often considered as a discrete analogue to convex functions; see [23]. Many combinatorial optimization problems can be formulated in terms of submodular functions.

Both minimizing and maximizing submodular functions, possibly under some extra constraints, have been considered extensively in the literature. Minimizing submodular functions can be performed efficiently with polynomially many oracle calls, either by the ellipsoid algorithm (see [12]) or through combinatorial algorithms that have been obtained in the last decade [29, 14, 15]. Unlike submodular function minimization, the problem of maximizing submodular functions is an NP-hard problem since it generalizes many NP-hard problems such as the maximum cut problem. In many settings, constant-factor approximation algorithms have been developed for this problem. Let us only mention that a $\frac{2}{5}$ -approximation has been developed for maximizing any non-negative, non-monotone submodular function [9], and that a $(1 - 1/e)$ -approximation algorithm has been derived for maximizing a monotone submodular function subject to a cardinality constraint [27], or an arbitrary matroid constraint [34]. Approximation algorithms for submodular analogues of several other well-known optimization problems have been studied, e.g., [35, 32].

Submodular functions have been of recent interest due to their applications in combinatorial auctions, particularly the submodular welfare problem [21, 18, 6]. This problem requires partitioning a set of items among a set of players in order to maximize their total utility. In this context, it is natural to assume that the players’ utility functions are submodular, as this captures a realistic notion of diminishing returns. Under this submodularity assumption, efficient approximation algorithms have recently been developed for this problem [6, 34].

*MIT Department of Mathematics. goemans@math.mit.edu. Supported by NSF contracts CCF-0515221 and CCF-0829878 and by ONR grant N00014-05-1-0148.

†Microsoft Research New England Lab, Cambridge, MA. nharvey@microsoft.com.

‡RIMS, Kyoto University, Kyoto 606-8502, Japan. iwata@kurims.kyoto-u.ac.jp. Supported by the Kayamori Foundation of Information Science Advancement.

§Google Research, New York, NY. mirrokni@gmail.com.

Contributions. The extensive literature on submodular functions motivates us to investigate other fundamental questions concerning their structure. How much information is contained in a submodular function? How much of that information can be obtained in just a few value oracle queries? Can an auctioneer efficiently estimate a player’s utility function if it is submodular? To address these questions, we consider the problem of approximating a submodular function f everywhere while performing only a polynomial number of queries. More precisely, the problem we study is:

Problem 1. Can one make $n^{O(1)}$ queries to f and construct a function \hat{f} (not necessarily submodular) which is an approximation of f , in the sense that $\hat{f}(S) \leq f(S) \leq g(n) \cdot \hat{f}(S)$ for all $S \subseteq [n]$. For what functions $g : \mathbb{N} \rightarrow \mathbb{R}$ is this possible?

For some submodular functions this problem can be solved exactly (i.e., with $g(n) = 1$). As an example, for graph cut functions, it is easy to see that one can completely reconstruct the graph in $O(n^2)$ queries. For more general submodular functions, we prove the following results.

- When f is a rank function of a matroid, we can compute a function \hat{f} after a polynomial number of queries giving an approximation factor $g(n) = \sqrt{n+1}$. Moreover, \hat{f} is submodular and has a particularly simple form: $\hat{f}(S) = \sqrt{\sum_{i \in S} c_i}$ for some $c \in \mathbb{R}_+^n$.
- When f is a general monotone submodular function, we can compute a submodular function \hat{f} after a polynomial number of queries giving an approximation factor $g(n) = O(\sqrt{n} \log n)$.
- On the other hand, we show that any algorithm performing a polynomial number of queries must satisfy $g(n) = \Omega(\sqrt{n}/\log n)$, even if f is the rank function of a matroid. If f is not necessarily monotone, we obtain the lower bound $g(n) = \Omega(\sqrt{n/\log n})$.

Related work. The lower bound mentioned above was previously described in an unpublished manuscript by M. Goemans, N. Harvey, R. Kleinberg and V. Mirrokni. This manuscript also gave a non-adaptive algorithm that solves Problem 1 when f is monotone with $g(n) = n/(c \log n)$ for any constant c ; furthermore, this is optimal (amongst non-adaptive algorithms).

A subsequent paper of Svitkina and Fleischer [32] considers several new optimization problems on submodular functions, as well as Problem 1. They give a randomized algorithm for Problem 1 that applies to a restricted class of submodular functions. Specifically, if there exists $R \subseteq [n]$ such that, for every $S \subseteq [n]$, the

value $f(S)$ depends only on $|S \cap R|$ and $|S \cap \bar{R}|$, then they can approximate f everywhere with $g(n) = 2\sqrt{n}$. Additionally, Svitkina and Fleischer adjusted the parameters of our lower bound construction, yielding an improved $\Omega(\sqrt{n/\log n})$ lower bound for Problem 1. They also show that this construction yields nearly-optimal lower bounds for several other problems that they consider.

Not only is our lower bound applicable to other submodular problems, but our algorithm is too. For example, it gives a deterministic $O(\sqrt{n} \log n)$ -approximation algorithm for the non-uniform submodular load balancing problem considered by Svitkina and Fleischer [32], by reducing it to load balancing on unrelated machines. This nearly matches the accuracy of their randomized $O(\sqrt{n} \log n)$ -approximation algorithm. As another example, we can reduce the submodular max-min fairness problem [11, 19] to the Santa Claus max-min fair allocation problem [2]. This yields an $O(n^{\frac{1}{2}} m^{\frac{1}{4}} \log n \log^{\frac{3}{2}} m)$ -approximation algorithm for the former problem where m is the number of buyers and n is the number of items. The existing algorithms for this problem obtain a $(n - m + 1)$ -approximation [11] and a $(2m - 1)$ -approximation [19]. These applications are discussed in Section 7.

Techniques. Our approximation results are based on ellipsoidal approximations to a centrally symmetric convex body K . An ellipsoid E constitutes a λ -ellipsoidal approximation of K if $E \subseteq K \subseteq \lambda E$. John’s theorem [16, p203] says that there always exists a \sqrt{n} -ellipsoidal approximation. We will elaborate on this fact in the following section.

One may also consider ellipsoidal approximations with an algorithmic view. When the body K is given by a separation oracle, it is known how to construct a $\sqrt{n(n+1)}$ -ellipsoidal approximation, using only a polynomial number of separation oracle calls. Details are in Grötschel, Lovász and Schrijver [12, p124]. Unfortunately, this general result is too weak for our purposes.

In our case, the convex body K is a symmetrized version of the polymatroid P_f associated with the monotone submodular function f , and we can exploit symmetries of this convex body. We show that a $(\sqrt{n+1}/\alpha)$ -ellipsoidal approximation is achievable for $\alpha \leq 1$, provided one can design a α^2 -approximation algorithm for the problem of maximizing a convex, separable, quadratic function over P_f . When f is the rank function of a matroid, this quadratic maximization problem can be solved easily and exactly in polynomial time (using the greedy algorithm), and this gives our $\sqrt{n+1}$ -approximation for rank functions of matroids. For general monotone submodular functions, the problem of maximizing (the square root of) a convex, separable,

quadratic function over a polymatroid P_f is equivalent to the Euclidean norm maximization problem (finding a vector of largest Euclidean norm) over a scaling of the polymatroid P_f . To tackle this latter problem, we proceed in two steps. We first show that a classical greedy algorithm provides a $(1 - 1/e)$ -approximation algorithm for the maximum Euclidean norm problem over a (*unscaled*) polymatroid P_g ; the analysis relies on the Nemhauser et al. [27] analysis of the greedy algorithm for maximizing a submodular function over a cardinality constraint. We then show that any scaled polymatroid Q can be approximated by a polymatroid P_g at a loss of a factor $O(\log n)$ (modulo a reasonable condition on the scaling): $\frac{1}{O(\log n)}P_g \subseteq Q \subseteq P_g$. This step involves properties of submodular functions (e.g., Lovász extensions) and polymatroids. Putting these pieces together, we get a $O(\sqrt{n} \log n)$ -approximation for any monotone submodular function everywhere.

2 Ellipsoidal Approximations

In this section, we state and review facts about ellipsoids, we discuss approximations of convex bodies by inscribed and circumscribed ellipsoids, and we build an algorithmic framework that we need for our approximation result. We focus on *centrally symmetric* convex bodies; in this case, one can exploit polarity to easily switch between inscribed and circumscribed ellipsoids.

In this paper, all matrices that we discuss are $n \times n$, real and symmetric. If a matrix A is positive definite we write $A \succ 0$, and if A is positive semi-definite we write $A \succeq 0$. Let $A \succ 0$ and let $A^{1/2}$ be its (unique) symmetric, positive definite square root: $A = A^{1/2}A^{1/2}$. We define the ellipsoidal norm $\|\cdot\|_A$ in \mathbb{R}^n by $\|x\|_A = \sqrt{x^T A x}$. Let B_n denote the (closed, Euclidean) unit ball $\{x \in \mathbb{R}^n : \|x\| \leq 1\}$, and let V_n denote its volume. Given $A \succ 0$, let $E(A)$ denote the ellipsoid (centered at the origin)

$$E(A) = \{x \in \mathbb{R}^n : x^T A x \leq 1\} = \{x : \|x\|_A \leq 1\}.$$

It is the image of the unit ball by a linear map: $E(A) = A^{-1/2}(B_n)$. The volume of $E(A)$ is $V_n/\det(A^{1/2})$. Given $c \in \mathbb{R}^n$, we have that

$$\max\{c^T x : x \in E(A)\} = \sqrt{c^T A^{-1} c} = \|c\|_{A^{-1}}.$$

Minimum volume circumscribed ellipsoid. Let K be a centrally symmetric ($x \in K$ iff $-x \in K$) convex body (compact convex set with non-empty interior) in \mathbb{R}^n . The minimum volume ellipsoid circumscribing K (i.e. containing K) is often referred to as the *Löwner ellipsoid* and can be formulated as a semi-infinite program:

$$(2.1) \quad \min\{-\log \det(A) : \begin{array}{l} \|x\|_A^2 \leq 1 \quad \forall x \in K \\ A \succ 0 \end{array} \}$$

where the variables are the symmetric matrix A . Observe that the constraints are linear in A : $\|x\|_A^2 = x^T A x \leq 1$. One of the main reasons for taking the log of the volume of the ellipsoid in the objective function is that the determinant of a matrix is strictly *log-concave* over positive definite matrices.

LEMMA 1. (FAN [8]¹) *Let $A, B \succ 0$, $A \neq B$, and $0 < \lambda < 1$. Then*

$$\log \det(\lambda A + (1 - \lambda)B) > \lambda \log \det A + (1 - \lambda) \log \det B.$$

The program (2.1) has therefore a strictly convex objective function (in A) and an infinite number of linear inequalities (in A), and is thus a “nice” convex program. In particular, we can solve the program efficiently provided we can separate over the constraints $\|x\|_A^2 \leq 1$. If the convex body K is polyhedral then we only need to write the constraints $\|x\|_A^2 \leq 1$ for its vertices since the maximization of a convex function $x^T A x$ (in x) over a polyhedral set K is always attained by a vertex. A case of particular interest is when K is defined as the convex hull of a given set of points [17, 20]; this case relates to optimal design problems in statistics.

The strict log-concavity of the determinant shows that the program (2.1) has a unique optimum solution, since a strict convex combination of any two distinct optimum solutions would give a strictly better solution. This shows that the minimum volume ellipsoid is unique, a result which is attributed to Löwner, and also follows from John’s proof [16].

Maximum volume inscribed ellipsoid. Using polarity, we can derive a similar formulation for the maximum volume ellipsoid inscribed in K (contained within K). For a convex body K , its polar K^* is defined as $\{c \in \mathbb{R}^n : c^T x \leq 1 \text{ for all } x \in K\}$. Observe that the polar of B_n is B_n itself, and, more generally, the polar of $E(A)$ is $E(A^{-1})$. Furthermore, for two convex bodies K and L , we have that $L \subseteq K$ iff $K^* \subseteq L^*$. Thus, the maximum volume ellipsoid $E(A)$ inscribed in K corresponds to the minimum volume ellipsoid $E(A^{-1})$ circumscribing K^* . The maximum volume inscribed ellipsoid is often called the *John ellipsoid*, although this attribution is somewhat inaccurate since John [16] actually considers only circumscribed ellipsoids. However, as remarked above, circumscribed and inscribed ellipsoids are interchangeable notions in the centrally symmetric case, so the inaccuracy is forgivable. The John ellipsoid $E(A)$ can be formulated by the following convex semi-infinite

¹Fan does not actually state the strict inequality, although his proof does show that it holds.

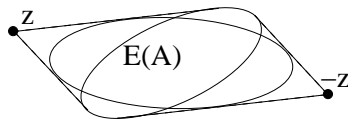
program polar to (2.1), which maximizes a concave function over a convex set.

$$\max\{ \log \det(A^{-1}) : \begin{array}{l} \|c\|_{A^{-1}}^2 \leq 1 \\ A^{-1} \succ 0 \end{array} \forall c \in K^* \}$$

Again, if K is polyhedral, we only need to write the constraint $\|c\|_{A^{-1}}^2 \leq 1$ for c such that $c^T x \leq 1$ defines a facet of K .

John's theorem. John's theorem, well-known in the theory of Banach spaces, says that K is contained in $\sqrt{n} \cdot E(A)$, where $E(A)$ is the maximum volume ellipsoid inscribed in K ; in other words, $\|x\|_A \leq \sqrt{n}$ for all $x \in K$. In terms of Banach spaces, this says that the (Banach-Mazur) distance between any n -dimensional Banach space (whose unit ball is K) and the n -dimensional Hilbert space l_2^n is at most \sqrt{n} .

John's theorem can be proved in several ways. See, for example, Ball [4] or Matoušek [24, §13.4]. We adopt a more algorithmic argument. Suppose there is an element $z \in K$ with $\|z\|_A > \sqrt{n}$. Then the following lemma gives an explicit construction of an ellipsoid of strictly larger volume that is contained in the convex hull of $E(A)$, z and $-z$, as illustrated in the figure. The resulting ellipsoid is larger since $k_n(l) > 1$ for $l > n$. This proves John's theorem.



LEMMA 2. For $A \succ 0$ and $z \in \mathbb{R}^n$ with $l = \|z\|_A^2 \geq n$, let

$$L(A, z) = \frac{n}{l} \frac{l-1}{n-1} A + \frac{n}{l^2} \left(1 - \frac{l-1}{n-1}\right) A z z^T A.$$

Then $L(A, z)$ is positive definite, the ellipsoid $E(L(A, z))$ is contained in $\text{conv}\{E(A), \{z, -z\}\}$, and its volume $\text{vol}(E(L(A, z)))$ equals $k_n(l) \cdot \text{vol}(E(A))$ where

$$k_n(l) = \sqrt{\left(\frac{l}{n}\right)^n \left(\frac{n-1}{l-1}\right)^{n-1}}.$$

In this extended abstract, most proofs are deferred to the full version. Actually, the lemma also follows from existing results by considering the polar statement, which says there exists an ellipsoid $E(B^{-1})$ containing

$$(2.2) \quad E(A^{-1}) \cap \{x : -1 \leq z^T x \leq 1\}$$

such that $\text{vol}(E(B^{-1})) < \text{vol}(E(A^{-1}))$, assuming $\|z\|_A > \sqrt{n}$. See, for example, Grötschel, Lovász and Schrijver [12, p72], Bland, Goldfarb and Todd [5, p1056], and Todd [33]. In fact, Todd derives

an expression for the minimum volume ellipsoid containing (2.2), which is precisely $B = L(A, z)$. This shows that $E(L(A, z))$ is indeed the John ellipsoid for $\text{conv}\{E(A), \{z, -z\}\}$.

3 Algorithm for Axis-Aligned Convex Bodies

In this section, we consider the question of constructing ellipsoidal approximations efficiently, we show how to exploit symmetries of the convex body, and we relate ellipsoidal approximations to the problem of approximating a submodular function everywhere.

We say that $E(A)$ is a λ -ellipsoidal approximation to K if $E(A) \subseteq K$ and $K \subseteq \lambda E(A)$. The John ellipsoid is therefore a \sqrt{n} -ellipsoidal approximation to a convex body K , and so is $1/\sqrt{n}$ times the Löwner ellipsoid. These are existential results. Algorithmically, the situation very much depends on how the convex body is given. If it is a polyhedral set given explicitly as the intersection of halfspaces then the convex program for the John's ellipsoid given above has one constraint for each given inequality and can be solved approximately, to within any desired accuracy. This gives an alternate way to derive the result of Grötschel, Lovász and Schrijver giving in polynomial-time a $\sqrt{n+1}$ -ellipsoidal approximation to a symmetric convex body K given explicitly by a system of linear inequalities. However, if K is given by a separation oracle and comes with the assumption of being well-bounded² then the best (known) algorithmic result is a polynomial-time algorithm giving only a $\sqrt{n(n+1)}$ -ellipsoidal approximation (Grötschel, Lovász, Schrijver [12], Theorem 4.6.3), and this will be too weak for our purpose. In fact, as was pointed out to us by José Soto, no algorithm, even randomized, can produce an approximation better than $\tilde{O}(n)$ for general centrally symmetric convex bodies.

The proof given above of John's theorem can be made algorithmic if we have an α -approximation algorithm ($\alpha \leq 1$) for maximizing $\|x\|_A$ over $x \in K$ and we are willing to settle for a $\sqrt{n+1}/\alpha$ -ellipsoidal approximation. In fact, we only need an α -approximate decision procedure which, given $A \succ 0$ with $E(A) \subseteq K$, either returns an $x \in K$ with $\|x\|_A > \sqrt{n+1}$ or guarantees that every $x \in K$ satisfies $\|x\|_A \leq \sqrt{n+1}/\alpha$. Assume we are given an ellipsoid $E_0 \subseteq K$ such that $K \subseteq pE_0$ (p is for example R/r in the definition of well-boundedness, and for our application, we will be able to use $p = n$). Iteratively, we find larger and larger (multiplicatively in volume) ellipsoids guaranteed to be within K . Given an ellipsoid $E_j = E(A_j)$ at iteration j ,

²As part of the input of this centrally symmetric convex body, we get $R \geq r > 0$ such that $B(0, r) \subseteq K \subseteq B(0, R)$, and the running time can thus involve $\log(R/r)$ polynomially.

suppose we run our α -approximate decision procedure for maximizing $\|x\|_A$ over K . Either (i) it returns a vector $z \in K$ with $\|z\|_{A_j} > \sqrt{n+1}$ or (ii) it guarantees that no $x \in K$ satisfies $\|x\|_{A_j} > \sqrt{n+1}/\alpha$. In case of (ii), we have a $\sqrt{n+1}/\alpha$ -ellipsoidal approximation. In case of (i), we can use Lemma 2 to find a larger ellipsoid $E_{j+1} = E(A_{j+1})$ also contained within K , and we can iterate. Our choice of the threshold $\sqrt{n+1}$ for the norm guarantees that $\text{vol}(E_{j+1})/\text{vol}(E_j) \geq 1 + \frac{1}{4n^2} - O(1/n^3)$, as stated in the lemma below. This increase in volume (and the fact that $K \subseteq pE_0$) guarantees that the number of iterations of this algorithm is at most $O(n^2 \log(p^n)) = O(n^3 \log p)$. One can get a smaller number of iterations with a higher threshold for the norm, see the Lemma below.

LEMMA 3. *For the function $k_n(l)$ given in Lemma 2, we have*

- $k_n(n+1) = 1 + \frac{1}{4n^2} - O(1/n^3)$,
- $k_n(2n) = \sqrt{2}e^{-1/4} - o(1) > 1$.

Ellipsoidal approximations for symmetrized polymatroids. Before we proceed, we describe the relationship between the problem of approximating a submodular function everywhere and these ellipsoidal approximations of centrally symmetric convex bodies.

For a monotone, submodular function $f : 2^{[n]} \rightarrow \mathbb{R}$ with $f(\emptyset) = 0$, its polymatroid $P_f \subseteq \mathbb{R}^n$ is defined by:

$$P(f) = \left\{ \begin{array}{l} x(S) \leq f(S), \quad \forall S \subseteq [n] \\ x \geq 0 \end{array} \right\}$$

where $x(S) = \sum_{i \in S} x_i$. To make it centrally symmetric, let $S(Q) = \{x \in \mathbb{R}^n : |x| \in Q\}$, where $|x|$ denotes component-wise absolute value. It is easy to see that, if $f(\{i\}) > 0$ for all i then $S(P_f)$ is a centrally symmetric convex body. (If there exists an index i with $f(\{i\}) = 0$, we can simply get rid of it as monotonicity and submodularity imply that $f(S) = f(S - i)$ for all S with $i \in S$.) Suppose now that $E(A)$ is a λ -ellipsoidal approximation to $S(P_f)$. This implies that, for any $c \in \mathbb{R}^n$,

$$\begin{aligned} \|c\|_{A^{-1}} &= \max\{c^T x : x \in E(A)\} \\ &\leq \max\{c^T x : x \in S(P_f)\} \\ &\leq \lambda \max\{c^T x : x \in E(A)\} = \lambda \|c\|_{A^{-1}}. \end{aligned}$$

In particular, taking $c = 1_S$ (the indicator vector for S) for any $S \subseteq [n]$, we get that

$$\|1_S\|_{A^{-1}} \leq f(S) \leq \lambda \|1_S\|_{A^{-1}},$$

where we have used the fact that $\max\{1_S^T x : x \in P_f\} = f(S)$. Thus the function \hat{f} defined by $\hat{f}(S) = \|1_S\|_{A^{-1}}$

provides a λ -approximation to $f(S)$ everywhere. In summary, a λ -ellipsoidal approximation to $S(P_f)$ gives a λ -approximation to $f(\cdot)$ everywhere.

Symmetry invariance. However, to be able to get a good ellipsoidal approximation, we need to exploit the symmetries of $S(P_f)$. Observe that if a centrally symmetric convex body K is invariant under a linear transformation T (i.e. $T(K) = K$) then, by uniqueness, the maximum volume inscribed ellipsoid E should also be invariant under T . More generally, define the automorphism group of K by $\text{Aut}(K) = \{T(x) = Cx : T(K) = K\}$. Then the maximum volume ellipsoid E inscribed in K satisfies $T(E) = E$ for all $T \in \text{Aut}(K)$, see for example [13]. In our case, $\text{Aut}(S(P_f))$ contains all transformations T of the form $T(x) = Cx$ where C is a diagonal ± 1 matrix. We call such convex bodies *axis aligned*. This means that the maximum volume ellipsoid $E(A)$ inscribed in $S(P_f)$ is also axis aligned, implying that A is a diagonal matrix.

Algorithm for axis-aligned convex bodies. Unfortunately, the algorithmic version of John's theorem presented above does not maintain axis-aligned ellipsoids. Indeed, for a diagonal matrix A , Lemma 2 does not produce an axis-aligned ellipsoid $E(L(A, z))$. However, we can use the following proposition to map $E(L(A, z))$ to an ellipsoid of no smaller volume (which shows that the maximum volume ellipsoid is axis aligned). We need some notation. For a vector $a \in \mathbb{R}^n$, let $\text{Diag}(a)$ be the diagonal matrix with main diagonal a ; for a matrix $A \in \mathbb{R}^{n \times n}$, let $\text{diag}(A) \in \mathbb{R}^n$ be its main diagonal.

PROPOSITION 3.1. *Let K be an axis-aligned convex body, and let $E(A)$ be an ellipsoid inscribed in K . Then the ellipsoid $E(B)$ defined by the diagonal matrix $B = (\text{Diag}(\text{diag}(A^{-1})))^{-1}$ satisfies (i) $E(B) \subseteq K$ and (ii) $\text{vol}(E(B)) \geq \text{vol}(E(A))$.*

(ii) is a restatement of Hadamard's inequality (applied to A^{-1}) which says that for a positive definite matrix C , $\det(C) \leq \prod_{i=1}^n c_{ii}$. To prove (i), one can show that $E(B) \subseteq \text{conv}\{T(E(A)) : T \in \text{Aut}(K)\}$.

Proposition 3.1 shows that, for an axis-aligned convex body such as $S(P_f)$, we can maintain throughout the algorithm axis-aligned ellipsoids. This has two important consequences. First, this means that we only need an α -approximate decision procedure for the case when A is diagonal. To emphasize this, we rename A by D . Recall that such a procedure, when given a $D \succ 0$ with $E(D) \subseteq S(P_f)$, either outputs a vector $x \in S(P_f)$ with $\|x\|_D > \sqrt{n+1}$ or guarantees that $\|x\|_D \leq \sqrt{n+1}/\alpha$ for all $x \in S(P_f)$. In section 4, we show that, for rank functions of matroids, $\max\{\|x\|_D :$

Algorithm AXIS-ALIGNED-ELLIPSOIDAL-APPROX

- ▷ Let $E_0 = E(D_0)$ be an axis-aligned ellipsoid inscribed in $S(P_f)$. One can choose $D_0 = \text{Diag}(v)$ where $v_i = n/f(\{i\})^2$ for $i \in [n]$.
- ▷ $j \leftarrow 0$.
- ▷ While $\text{MAX-NORM}(D)$ returns a vector z with $\|z\|_{D_j} > \sqrt{n+1}$ do
 - ▷ $B \leftarrow L(D_j, z)$
 - ▷ $D_{j+1} \leftarrow (\text{Diag}(\text{diag}(B^{-1})))^{-1}$
 - ▷ $j \leftarrow j + 1$
- ▷ Return the function \hat{f} given by $\hat{f}(S) = \sqrt{\sum_{i \in S} p_i}$ where $p = \text{diag}(D_j^{-1})$.

Figure 1: The algorithm for constructing a function \hat{f} which is a $\sqrt{n+1}/\alpha$ -approximation to f .

$x \in S(P_f)$ can be solved exactly (thus $\alpha = 1$) and efficiently (in polynomial time and with polynomially many oracle calls), while in Section 5, we describe an efficient $1/O(\log n)$ -decision procedure for general monotone submodular functions. Secondly, the function \hat{f} we construct based on an ellipsoidal approximation takes a particularly simple form when the ellipsoid $E(D)$ is given by a diagonal matrix D . In this case, $\hat{f}(S) = \|1_S\|_{D^{-1}}$ reduces to:

$$\hat{f}(S) = \sqrt{\sum_{i \in S} p_i},$$

where $p_i = 1/D_{ii}$ for $i \in [n]$. Observe that this approximation \hat{f} is actually *submodular* (while this was not necessarily the case for non axis-aligned ellipsoids).

Summarizing, Figure 1 gives our algorithm for constructing a $\sqrt{n+1}/\alpha$ -ellipsoidal approximation of $S(P_f)$ and thus a $\sqrt{n+1}/\alpha$ -approximation to f everywhere, given an α -approximate decision procedure $\text{MAX-NORM}(D)$ for maximizing $\|x\|_D$ over $S(P_f)$ (or equivalently over P_f , by symmetry) for a positive definite *diagonal* matrix D (i.e. $d_{ii} > 0$).

One can easily check that the ellipsoid $E_0 = E(D_0)$ given in the algorithm is an n -ellipsoidal approximation: it satisfies $E_0 \subseteq S(P_f)$ and $S(P_f) \subseteq nE_0$.

THEOREM 4. *If $\text{MAX-NORM}(D)$ is an α -approximate decision procedure for $\max\{\|x\|_D : x \in P_f\}$ then AXIS-ALIGNED-ELLIPSOIDAL-APPROX outputs a $\sqrt{n+1}/\alpha$ -approximation to f everywhere after at most $O(n^3 \log n)$ iterations.*

4 Matroid Rank Functions

Let $M = ([n], \mathcal{I})$ be a matroid and \mathcal{I} its family of independent sets. Let $f(\cdot)$ be its rank function: $f(S) = \max\{|U| : U \subseteq S, U \in \mathcal{I}\}$ for $S \subseteq [n]$. f is monotone and submodular and the corresponding polymatroid P_f is precisely the convex hull of characteristic vectors of independent sets (Edmonds [7]).

For a matroid rank function f , the problem $\max\{\|x\|_D : x \in P_f\}$ can be solved exactly in polynomial-time and with a polynomial number of oracle calls, when D is a positive definite, diagonal matrix. Indeed, maximizing $\|x\|_D$ is equivalent to maximizing its square: $\max\{\sum_i d_i x_i^2 : x \in P_f\}$, where $d = \text{diag}(D)$. This is the maximization of a convex function over a polyhedral set, and therefore the maximum is attained at one of the vertices. But any vertex x of P_f is a 0-1 vector [7] and thus satisfies $x_i^2 = x_i$. The problem is thus equivalent to maximizing the linear function $\sum_i d_i x_i$ over P_f which can be solved in polynomial-time by the greedy algorithm for finding a maximum weight independent set in a matroid. Therefore, AXIS-ALIGNED-ELLIPSOIDAL-APPROX gives a $\sqrt{n+1}$ -approximation everywhere for rank functions of matroids.

We should emphasize that the simple approach of linearizing x_i^2 by x_i would have failed if our ellipsoids were not axis aligned, i.e., if D were not diagonal. In fact, the *quadratic spanning tree problem*, defined as $\max\{\|x\|_D : x \in P_f\}$ where P_f is a graphic matroid polytope and D is a symmetric, non-diagonal matrix, is NP-hard as it includes the Hamiltonian path problem as a special case [3]. We remark that NP-hardness holds even if D is positive definite.

5 General Monotone Submodular Functions

In this section, we present a $1/O(\log n)$ -approximate decision procedure for $\max\{\|x\|_D : x \in P_f\}$ for a general monotone submodular function f . Taking squares, we rewrite the problem as:

$$(5.3) \quad \max \left\{ \sum_{i=1}^n c_i^2 x_i^2 : x \in P_f \right\},$$

where we let $c = \text{diag}(D^{1/2})$. Assuming that the ellipsoid $E(D)$ is inscribed in $S(P_f)$, we will either find an $x \in P_f$ for which $\sum_{i=1}^n c_i^2 x_i^2 > n+1$ or guarantee that no $x \in P_f$ gives a value greater than $(n+1)/\alpha^2$,

where $\alpha = 1/O(\log n)$.

We first consider the case in which all $c_i = 1$, and derive a $(1 - 1/e)^2$ -approximation algorithm for (5.3). Consider the following greedy algorithm. Let $T_0 = \emptyset$, and for every $k = 1, \dots, n$, let

$$T_k = \arg \max_{S=T_{k-1} \cup \{j\}, j \notin T_{k-1}} f(S),$$

that is, we repeatedly add the element which gives the largest increase in the submodular function value. Let $\hat{x} \in P_f$ be the vector defined by $\hat{x}(T_k) = f(T_k)$ for $1 \leq k \leq n$; the fact that \hat{x} is in P_f is a fundamental property of polymatroids. We claim that \hat{x} provides a $(1 - 1/e)^2$ -approximation for (5.3) when all c_i 's are 1.

LEMMA 5. *For the solution \hat{x} constructed above, we have*

$$\sum_{i=1}^n \hat{x}_i^2 \geq \left(1 - \frac{1}{e}\right)^2 \max \left\{ \sum_{i=1}^n x_i^2 : x \in P_f \right\}.$$

PROOF. Nemhauser, Wolsey and Fisher [27] show that, for every $k \in [n]$, we have

$$f(T_k) \geq \left(1 - \frac{1}{e}\right) \max_{S:|S|=k} f(S).$$

Let $h(k) = f(T_k)$ for $k \in [n]$; because of our greedy choice and submodularity of f , $h(\cdot)$ is concave. Define the monotone submodular function ℓ by $\ell(S) = \frac{e}{e-1} h(|S|)$. The fact that ℓ is submodular comes from the concavity of h . Observe that, for every S , $f(S) \leq \ell(S)$, and therefore, $P_f \subseteq P_\ell$ and

$$\max \left\{ \sum_{i=1}^n x_i^2 : x \in P_f \right\} \leq \max \left\{ \sum_{i=1}^n x_i^2 : x \in P_\ell \right\}.$$

By convexity of the objective function, the maximum over P_ℓ is attained at a vertex. But all vertices of P_ℓ are permutations of the coordinates of $\frac{e}{e-1} \hat{x}$ (or are dominated by such vertices), and thus

$$\max \left\{ \sum_{i=1}^n x_i^2 : x \in P_f \right\} \leq \left(\frac{e}{e-1}\right)^2 \left(\sum_{i=1}^n \hat{x}_i^2\right).$$

□

We now deal with the case when the c_i 's are arbitrary. First our guarantee that the ellipsoid $E(D)$ is within $S(P_f)$ means that $f(\{i\})e_i$ (where e_i is the i th unit vector) is not in the interior of $E(D)$, i.e. we must have $c_i f(\{i\}) \geq 1$ for all $i \in [n]$. We can also assume that $c_i f(\{i\}) \leq \sqrt{n+1}$. If not, $x = f(\{i\})e_i$ constitutes a vector in P_f with $\sum_j c_j^2 x_j^2 > n+1$. Thus, for all $i \in [n]$, we can assume that $1 \leq c_i f(\{i\}) \leq \sqrt{n+1}$.

To reduce to the case with $c_i = 1$ for all i , consider the linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^n : x \rightarrow y = (c_1 x_1, \dots, c_n x_n)$. The problem $\max\{\sum_i c_i^2 x_i^2 : x \in P_f\}$ is equivalent to $\max\{\sum_i y_i^2 : y \in T(P_f)\}$. Unfortunately, $T(P_f)$ is not a polymatroid, but it is contained in the polymatroid P_g defined by:

$$\begin{aligned} g(S) &= \max \left\{ \sum_{i \in S} y_i : y \in T(P_f) \right\} \\ &= \max \left\{ \sum_{i \in S} c_i x_i : x \in P_f \right\}. \end{aligned}$$

The fact that g is submodular can be derived either from first principles (exploiting the correctness of the greedy algorithm) or as follows. The Lovász extension \hat{f} of f is defined as $\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R} : w \rightarrow \max\{w^T x : x \in P_f\}$ (see Lovász [23] or [10]). It is L -convex, see Murota [25, Prop. 7.25], meaning that, for $w_1, w_2 \in \mathbb{R}^n$, $\hat{f}(w_1) + \hat{f}(w_2) \geq \hat{f}(w_1 \vee w_2) + \hat{f}(w_1 \wedge w_2)$, where \vee (resp. \wedge) denotes component-wise max (resp. min). The submodularity of g now follows from the L -convexity of \hat{f} by taking vectors w obtained from c by zeroing out some coordinates.

We can approximately (within a factor $(1 - 1/e)^2$) compute $\max\{\sum_i y_i^2 : y \in P_g\}$, or equivalently approximate $\max\{\sum_i c_i^2 x_i^2 : x \in T^{-1}(P_g)\}$. The question is how much "bigger" is $T^{-1}(P_g)$ compared to P_f ? To answer this question, we perform another polymatroidal approximation, this time of $T^{-1}(P_g)$ and define the submodular function h by:

$$\begin{aligned} h(S) &= \max \left\{ \sum_{i \in S} x_i : x \in T^{-1}(P_g) \right\} \\ &= \max \left\{ \sum_{i \in S} \frac{1}{c_i} y_i : y \in P_g \right\}. \end{aligned}$$

Again, $h(\cdot)$ is submodular and we can easily obtain a closed form expression for it, see Lemma 8. We have thus sandwiched $T^{-1}(P_g)$ between P_f and P_h : $P_f \subseteq T^{-1}(P_g) \subseteq P_h$. To show that all these polytopes are close to each other, we show the following theorem whose proof is deferred to the full version:

THEOREM 6. *Suppose that for all $i \in [n]$, we have $1 \leq c_i f(\{i\}) \leq \sqrt{n+1}$. Then, for all $S \subseteq [n]$, $h(S) \leq (2 + \frac{3}{2} \ln(n)) f(S)$.*

Our algorithm is now the following. Using the $(1 - 1/e)^2$ -approximation algorithm applied to P_g , we find a vector $\hat{x} \in T^{-1}(P_g)$ such that

$$\sum_i c_i^2 \hat{x}_i^2 \geq \left(1 - \frac{1}{e}\right)^2 \max \left\{ \sum_i c_i^2 x_i^2 : x \in T^{-1}(P_g) \right\}.$$

Now, by Theorem 6, we know that $\tilde{x} = \hat{x}/O(\log n)$ is in P_f . Therefore, we have that

$$\begin{aligned} \sum_i c_i^2 \tilde{x}_i^2 &= \frac{1}{O(\log^2(n))} \sum_i c_i^2 \hat{x}_i^2 \\ &\geq \frac{1}{O(\log^2(n))} \max \{ \sum_i c_i^2 x_i^2 : x \in T^{-1}(P_g) \} \\ &\geq \frac{1}{O(\log^2(n))} \max \{ \sum_i c_i^2 x_i^2 : x \in P_f \}, \end{aligned}$$

giving us the required approximation guarantee.

The lemmas below give a closed form expression for $g(\cdot)$ and $h(\cdot)$; their proofs are used in the proof of Theorem 6. They follow from the fact that the greedy algorithm can be used to maximize a linear function over a polymatroid. Both lemmas apply to any set S after renumbering its indices. For any i and j , we define $[i, j] = \{k \in \mathbb{N} : i \leq k \leq j\}$ and $f(i, j) = f([i, j])$. Observe that $f(i, j) = 0$ for $i > j$.

LEMMA 7. For $S = [k]$ with $c_1 \leq c_2 \leq \dots \leq c_k$, we have $g(S) = \sum_{i=1}^k c_i [f(i, k) - f(i+1, k)]$.

LEMMA 8. For $S = [k]$ with $c_1 \leq \dots \leq c_k$, we have:

$$\begin{aligned} h(S) &= \sum_{i,j: 1 \leq i \leq j \leq k} \frac{c_i}{c_j} \cdot \left(f(i, j) - f(i+1, j) \right. \\ &\quad \left. - f(i, j-1) + f(i+1, j-1) \right) \\ &= \sum_{l,m: 1 \leq l \leq m \leq k} (c_l - c_{l-1}) \left(\frac{1}{c_m} - \frac{1}{c_{m+1}} \right) f(l, m). \end{aligned}$$

6 Lower Bound

In this section, we show that approximating a submodular function everywhere requires an approximation ratio of $\Omega(\sqrt{n}/\log n)$, even when restricting f to be a matroid rank function (and hence monotone). For non-monotone submodular functions, we show that the approximation ratio must be $\Omega(\sqrt{n}/\log n)$.

The argument has two steps:

- **Step 1.** Construct a family of submodular functions parameterized by natural numbers $\alpha > \beta$ and a set $R \subseteq [n]$ which is unknown to the algorithm.
- **Step 2.** Use discrepancy arguments to determine whether a sequence of queries can determine R .

This analysis leads to a choice of α and β .

Step 1. Let \mathbf{U} be the uniform rank- α matroid on $[n]$; its rank function is

$$r_{\mathbf{U}}(S) = \min \{|S|, \alpha\}.$$

Now let $R \subseteq [n]$ be arbitrary such that $|R| = \alpha$. We define a matroid \mathbf{M}_R by letting its independent sets be

$$\mathcal{I}_{\mathbf{M}_R} = \{I \subseteq [n] : |I| \leq \alpha \text{ and } |I \cap R| \leq \beta\}.$$

This matroid can be viewed as a partition matroid, truncated to rank α . One can check that its rank function is

$$r_{\mathbf{M}_R}(S) = \min \{|S|, \beta + |S \cap \bar{R}|, \alpha\}.$$

Now we consider when $r_{\mathbf{U}}(S) \neq r_{\mathbf{M}_R}(S)$. By the equations above, it is clear that this holds iff

$$(6.4) \quad \beta + |S \cap \bar{R}| < \min \{|S|, \alpha\}.$$

Case 1: $|S| \leq \alpha$. Eq. (6.4) holds iff $\beta + |S \cap \bar{R}| < |S|$, which holds iff $\beta < |S \cap R|$. That inequality together with $|S| \leq \alpha$ implies that $|S \cap \bar{R}| < \alpha - \beta$.

Case 2: $|S| > \alpha$. Eq. (6.4) holds iff $\beta + |S \cap \bar{R}| < \alpha$. That inequality implies that $|S \cap R| > \beta + (|S| - \alpha) > \beta$.

Our family of monotone functions is

$$\mathcal{F} = \{r_{\mathbf{M}_R} : R \subseteq [n], |R| = \alpha\} \cup \{r_{\mathbf{U}}\}.$$

Our family of non-monotone functions is

$$\mathcal{F}' = \{r_{\mathbf{M}_R} + h : R \subseteq [n], |R| = \alpha\} \cup \{r_{\mathbf{U}} + h\},$$

where h is the function defined by $h(S) = -|S|/2$.

Step 2 (Non-monotone case). Consider any algorithm which is given a function $f \in \mathcal{F}'$, performs a sequence of queries $f(S_1), \dots, f(S_k)$, and must distinguish whether $f = r_{\mathbf{U}} + h$ or $f = r_{\mathbf{M}_R} + h$ (for some R). For the sake of distinguishing these possibilities, the added function h is clearly irrelevant; it only affects the approximation ratio. By our discussion above, the algorithm can distinguish $r_{\mathbf{M}_R}$ from $r_{\mathbf{U}}$ only if one of the following two cases occurs.

Case 1: $\exists i$ such that $|S_i| \leq \alpha$ and $|S_i \cap R| > \beta$.

Case 2: $\exists i$ such that $|S_i| > \alpha$ and $\beta + |S_i \cap \bar{R}| < \alpha$.

As argued above, if *either* of these cases hold then we have *both* $|S_i \cap R| > \beta$ and $|S_i \cap \bar{R}| < \alpha - \beta$. Thus

$$(6.5) \quad |S_i \cap R| - |S_i \cap \bar{R}| > 2\beta - \alpha.$$

Now consider the family of sets $\mathcal{A} = \{S_1, \dots, S_k, [n]\}$. A standard result [1, Theorem 12.1.1] on the discrepancy of \mathcal{A} shows that there exists an R such that

$$(6.6a) \quad \left| |S_i \cap R| - |S_i \cap \bar{R}| \right| \leq \epsilon \quad \forall i$$

$$(6.6b) \quad \left| |[n] \cap R| - |[n] \cap \bar{R}| \right| \leq \epsilon,$$

where $\epsilon = \sqrt{2n \ln(2k)}$. Eq. (6.6b) implies that $|R| = n/2 + \epsilon'$, where $|\epsilon'| \leq \epsilon/2$. By definition, $\alpha = |R|$. So if we choose $\beta = n/4 + \epsilon$ then $2\beta - \alpha > \epsilon$. Thus Eq. (6.5) cannot hold, since it would contradict Eq. (6.6a). This shows that the algorithm cannot distinguish $f = r_{\mathbf{M}_R} + h$ from $f = r_{\mathbf{U}} + h$.

The approximation ratio of the algorithm is at most $f'(R)/f(R)$. We have $f'(R) = |R| - |R|/2 = |R|/2$ and $f(R) = \beta - |R|/2 \leq (n/4 + \epsilon) - (n/2 - \epsilon)/2 < 2\epsilon$. This shows that no deterministic algorithm can achieve approximation ratio better than

$$\frac{f'(R)}{f(R)} = \frac{|R|}{4\epsilon} \geq \frac{n/2 - \epsilon}{4\epsilon} = \Omega(\sqrt{n/\log k})$$

Since $k = n^{O(1)}$, this proves the claimed result. If $k = O(n)$ then the lower bound improves to $\Omega(\sqrt{n})$ via a result of Spencer [30].

The construction of the set R in [1, Theorem 12.1.1] is probabilistic: choosing R uniformly at random works with high probability, regardless of the algorithm's queries S_1, \dots, S_k . This implies that the lower bound also applies to randomized algorithms.

Step 2 (Monotone case). In this case, we pick $\alpha \approx \sqrt{n}$ and $\beta = \Omega(\ln k)$. The argument is similar to the non-monotone case except that we cannot apply standard discrepancy results since they do not construct R with $|R| = \alpha \approx \sqrt{n}$. Instead, we derive analogous results using Chernoff bounds. We construct R by picking each element independently with probability $1/\sqrt{n}$. With high probability $|R| = \Theta(\sqrt{n})$. We must now bound the probability that the algorithm succeeds.

Case 1: Given $|S_i| \leq \alpha$, what is $\Pr[|S_i \cap R| > \beta]$? We have $\mathbb{E}[|R \cap S_i|] = |S_i|/\sqrt{n} = O(1)$. Chernoff bounds show that $\Pr[|R \cap S_i| > \beta] \leq \exp(-\beta/2) = 1/k^2$.

Case 2: Given $|S_i| > \alpha$, what is $\Pr[\beta + |S_i \cap \bar{R}| < \alpha]$? As observed above, this event is equivalent to $|S_i \cap R| > \beta + (|S_i| - \alpha) =: \xi$. Let $\mu = \mathbb{E}[|S_i \cap \bar{R}|] = |S_i|/\sqrt{n}$. Note that

$$\frac{\xi}{\mu} = \frac{\log n}{|S_i|/\sqrt{n}} + \sqrt{n} \cdot \left(1 - \frac{\alpha}{|S_i|}\right),$$

which is $\Omega(\log n)$ for any value of $|S_i|$. A Chernoff bound then shows that $\Pr[|S_i \cap R| > \xi] < \exp(-\xi/2) \leq 1/k^2$.

A union bound shows that none of these events occur with high probability, and thus the algorithm fails to distinguish r_{M_R} from r_U . The approximation ratio of the algorithm is at most $f'(R)/f(R) = \alpha/\beta = \Omega(\sqrt{n}/\log k)$. This lower bound also applies to randomized algorithms, by the same reasoning as in the non-monotone case. Since $k = n^{O(1)}$, this proves the desired result.

7 Applications

7.1 Submodular Load Balancing

Let f_1, \dots, f_m be monotone submodular functions on the ground set $[n]$. The non-uniform submodular load balancing problem is

$$(7.7) \quad \min_{V_1, \dots, V_m} \max_j f_j(V_j),$$

where the minimization is over partitions of $[n]$ into V_1, \dots, V_m .

Suppose we construct the approximations $\hat{f}_1, \dots, \hat{f}_m$ such that

$$\hat{f}_j(S) \leq f_j(S) \leq g(n) \cdot \hat{f}_j(S) \quad \forall j \in [m], S \subseteq [n].$$

Furthermore, suppose that each \hat{f}_j is of the form

$$\hat{f}_j(S) = \sqrt{\sum_{i \in S} c_{j,i}},$$

for some non-negative real values $c_{j,i}$. Consider the problem of finding a partition V_1, \dots, V_m that minimizes $\max_j \hat{f}_j(V_j)$. By squaring, we would like to solve

$$(7.8) \quad \min_{V_1, \dots, V_m} \max_j \sum_{i \in V_j} c_{j,i}.$$

This is precisely the problem of scheduling jobs without preemption on non-identical parallel machines, while minimizing the makespan. In deterministic polynomial time, one can compute a 2-approximate solution X_1, \dots, X_m to this problem [22], which also gives an approximate solution to Eq. (7.7).

Formally, let W_1, \dots, W_m be an optimal solution to Eq. (7.8), let X_1, \dots, X_m be a solution computed using the algorithm of [22], and let Y_1, \dots, Y_m be an optimal solution to the original problem in Eq. (7.7). Then we have $\frac{1}{2} \cdot \max_j \hat{f}_j^2(X_j) \leq \max_j \hat{f}_j^2(W_j)$, and thus

$$\frac{1}{\sqrt{2}g(n)} \cdot \max_j f_j(X_j) \leq \max_j f_j(Y_j).$$

Thus, the X_j 's give a $(\sqrt{2}g(n))$ -approximate solution to Eq. (7.7). Applying the algorithm of Section 5 to construct the \hat{f}_j 's, we obtain an $O(\sqrt{n} \log n)$ -approximation to the non-uniform submodular load balancing problem.

7.2 Submodular Max-Min Fair Allocation

Consider m buyers and a ground set $[n]$ of items. Let f_1, \dots, f_m be monotone submodular functions on the ground set $[n]$, and let f_j be the valuation function of buyer j . The submodular max-min fair allocation problem is

$$(7.9) \quad \max_{V_1, \dots, V_m} \min_j f_j(V_j),$$

where the maximization is over partitions of $[n]$ into V_1, \dots, V_m . This problem was studied by Golovin [11] and Khot and Ponnuswami [19]. Those papers respectively give algorithms achieving an $(n - m + 1)$ -approximation and a $(2m - 1)$ -approximation. Here we

give a $O(n^{\frac{1}{2}}m^{\frac{1}{4}}\log n\log^{\frac{3}{2}}m)$ -approximation algorithm for this problem.

The idea of the algorithm is similar to that of the load balancing problem. We construct the approximations $\hat{f}_1, \dots, \hat{f}_m$

$$\hat{f}_j(S) \leq f_j(S) \leq g(n) \cdot \hat{f}_j(S) \quad \forall j \in [m], S \subseteq [n],$$

such that \hat{f}_j is of the form

$$\hat{f}_j(S) = \sqrt{\sum_{i \in S} c_{j,i}},$$

for some non-negative real values $c_{j,i}$. Consider the problem of finding a partition V_1, \dots, V_m that maximizes $\min_j \hat{f}_j(V_j)$. By squaring, we would like to solve

$$\max_{V_1, \dots, V_m} \min_j \sum_{i \in V_j} c_{j,i}.$$

This problem is the Santa Claus max-min fair allocation problem, for which Asadpour and Saberi [2] give a $O(\sqrt{m}\log^3 m)$ approximation algorithm. Using this, together with the algorithm of Section 5 to construct the \hat{f}_j 's, we obtain an $O(n^{\frac{1}{2}}m^{\frac{1}{4}}\log n\log^{\frac{3}{2}}m)$ -approximation for the submodular max-min fair allocation problem.

Acknowledgements

The authors thank Robert Kleinberg for helpful discussions at a preliminary stage of this work, José Soto for discussions on inertial ellipsoids, and Uri Feige for his help with the analysis of Section 6.

References

- [1] N. Alon and J. Spencer. "The Probabilistic Method". Wiley, second edition, 2000.
- [2] A. Asadpour and A. Saberi. "An approximation algorithm for max-min fair allocation of indivisible goods". *STOC*, 114–121, 2007.
- [3] A. Assad and W. Xu. "The Quadratic Minimum Spanning Tree Problem". *Naval Research Logistics*, **39**, 1992.
- [4] K. Ball. "An Elementary Introduction to Modern Convex Geometry". *Flavors of Geometry*, MSRI Publications, 1997.
- [5] R. G. Bland, D. Goldfarb and M. J. Todd. "The Ellipsoid Method: A Survey". *Operations Research*, **29**, 1981.
- [6] S. Dobzinski and M. Schapira. "An improved approximation algorithm for combinatorial auctions with submodular bidders". *SODA*, 1064–1073, 2006.
- [7] J. Edmonds, "Matroids and the Greedy Algorithm", *Mathematical Programming*, **1**, 127–136, 1971.
- [8] K. Fan, "On a theorem of Weyl concerning the eigenvalues of linear transformations, II", *Proc. Nat. Acad. Sci.*, 1950.
- [9] U. Feige, V. Mirrokni and J. Vondrák, "Maximizing non-monotone submodular functions", *FOCS*, 461–471, 2007.
- [10] S. Fujishige, "Submodular Functions and Optimization", volume 58 of *Annals of Discrete Mathematics*. Elsevier, second edition, 2005.

- [11] D. Golovin, "Max-Min Fair Allocation of Indivisible Goods". Technical Report CMU-CS-05-144, 2005.
- [12] M. Grötschel, L. Lovász, and A. Schrijver, "Geometric Algorithms and Combinatorial Optimization", Springer Verlag, second edition, 1993.
- [13] O. Güler and F. Gürtina, "The extremal volume ellipsoids of convex bodies, their symmetry properties, and their determination in some special cases", arXiv:0709.707v1.
- [14] S. Iwata, L. Fleischer, and S. Fujishige, "A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions", *Journal of the ACM*, **48**, 761–777, 2001.
- [15] S. Iwata and J. Orlin, "A Simple Combinatorial Algorithm for Submodular Function Minimization", *SODA*, 2009.
- [16] F. John. "Extremum problems with inequalities as subsidiary conditions", *Studies and Essays, presented to R. Courant on his 60th Birthday, January 8, 1948*, Interscience, New York, 187–204, 1948.
- [17] L. G. Khachiyan. "Rounding of polytopes in the real number model of computation", *Math of OR*, **21**, 307–320, 1996.
- [18] S. Khot, R. Lipton, E. Markakis and A. Mehta. "Inapproximability results for combinatorial auctions with submodular utility functions", *WINE*, 92–101, 2005.
- [19] S. Khot and A. Ponnuswami. "Approximation Algorithms for the Max-Min Allocation Problem". *APPROX-RANDOM*, 204–217, 2007.
- [20] P. Kumar and E. A. Yildirim, "Minimum-Volume Enclosing Ellipsoids and Core Sets", *Journal of Optimization Theory and Applications*, **126**, 1–21, 2005.
- [21] B. Lehmann, D. J. Lehmann and N. Nisan. "Combinatorial auctions with decreasing marginal utilities", *Games and Economic Behavior*, **55**, 270–296, 2006.
- [22] J. K. Lenstra, D. B. Shmoys and E. Tardos. "Approximation algorithms for scheduling unrelated parallel machines". *Mathematical Programming*, **46**, 259–271, 1990.
- [23] L. Lovász, "Submodular Functions and Convexity", in A. Bachem et al., eds, *Mathematical Programming: The State of the Art*, 235–257, 1983.
- [24] J. Matoušek, "Lectures on Discrete Geometry". Springer, 2002.
- [25] K. Murota, "Discrete Convex Analysis", SIAM Monographs on Discrete Mathematics and Applications, SIAM, 2003.
- [26] H. Narayanan, "Submodular Functions and Electrical Networks", Elsevier, 1997.
- [27] G. L. Nemhauser, L. A. Wolsey and M. L. Fisher. "An analysis of approximations for maximizing submodular set functions I". *Mathematical Programming*, **14**, 1978.
- [28] A. Schrijver, "Combinatorial Optimization: Polyhedra and Efficiency". Springer, 2004.
- [29] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time", *Journal of Combinatorial Theory, Series B*, **80**, 346–355, 2000.
- [30] J. Spencer, "Six Standard Deviations Suffice", *Trans. Amer. Math. Soc.*, **289**, 679–706, 1985.
- [31] P. Sun and R. M. Freund. "Computation of Minimum Volume Covering Ellipsoids", *Operations Research*, **52**, 690–706, 2004.
- [32] Z. Svitkina and L. Fleischer. "Submodular Approximation: Sampling-Based Algorithms and Lower Bounds". *FOCS*, 2008.
- [33] M. J. Todd. "On Minimum Volume Ellipsoids Containing Part of a Given Ellipsoid". *Math of OR*, 1982.
- [34] J. Vondrák. "Optimal Approximation for the Submodular Welfare Problem in the Value Oracle Model". *STOC*, 2008.
- [35] L. A. Wolsey. "An Analysis of the Greedy Algorithm for the Submodular Set Covering Problem". *Combinatorica*, **2**, 385–393, 1982.