Symmetry and approximability of submodular maximization problems

Jan Vondrák*
IBM Almaden Research Center
San Jose, CA 95120
Email: jvondrak@us.ibm.com

Abstract— A number of recent results on optimization problems involving submodular functions have made use of the "multilinear relaxation" of the problem [3], [8], [24], [14], [13]. We present a general approach to deriving inapproximability results in the value oracle model, based on the notion of "symmetry gap". Our main result is that for any fixed instance that exhibits a certain "symmetry gap" in its multilinear relaxation, there is a naturally related class of instances for which a better approximation factor than the symmetry gap would require exponentially many oracle queries.

This unifies several known hardness results for submodular maximization, e.g. the optimality of $(1-1/e)$-approximation for monotone submodular maximization under a cardinality constraint [20], [7], and the impossibility of $(\frac{1}{2}+\epsilon)$-approximation for unconstrained (non-monotone) submodular maximization [8]. It follows from our result that $(\frac{1}{2}+\epsilon)$-approximation is also impossible for non-monotone submodular maximization subject to a (non-trivial) matroid constraint. On the algorithmic side, we present a 0.309-approximation for this problem, improving the previously known factor of $\frac{1}{4}-o(1)$ [14].

As another application, we consider the problem of maximizing a non-monotone submodular function over the bases of a matroid. A $(\frac{1}{6}-o(1))$-approximation has been developed for this problem, assuming that the matroid contains two disjoint bases [14]. We show that the best approximation one can achieve is indeed related to packings of bases in the matroid. Specifically, for any $k \geq 2$, there is a class of matroids of fractional base packing number $\nu = \frac{k}{k-1}$, such that any algorithm achieving a better than $(1-\frac{1}{\nu})$-approximation for this class would require exponentially many value queries. On the positive side, we present a $\frac{1}{2}(1-\frac{1}{\nu}-o(1))$-approximation algorithm for the same problem.

Our hardness results hold in fact for very special *symmetric instances*. For such symmetric instances, we show that the approximation factors of $\frac{1}{2}$ (for submodular maximization subject to a matroid constraint) and $1-\frac{1}{\nu}$ (for a matroid base constraint) can be achieved algorithmically and hence are optimal.

Keywords-approximation algorithms; submodular functions; matroids; multilinear extension

1. Introduction

Submodular set functions are defined by the following condition for all pairs of sets S, T:

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

or equivalently by the property that the *marginal value* of any element, $f_S(j) = f(S+j) - f(S)$, satisfies $f_T(j) \leq f_S(j)$, whenever $j \notin T \supset S$. In addition, a set function is called monotone if $f(S) \leq f(T)$ whenever $S \subseteq T$. Throughout this paper, we assume that $f(S)$ is nonnegative.

Submodular functions have been studied in the context of combinatorial optimization since the 1970's, especially in connection with matroids [5], [6], [18], [19], [20], [26], [27], [16], [10]. Submodular functions appear mostly for the following two reasons: (i) submodularity arises naturally in various combinatorial settings, and many algorithmic applications use it either explicitly or implicitly; (ii) submodularity has a natural interpretation as the property of *diminishing returns*, which defines an important class of utility/valuation functions. Submodularity as an abstract concept is both general enough to be useful for applications and it carries enough structure to allow strong positive results. A fundamental algorithmic result is that any submodular function given by a value oracle can be *minimized* in strongly polynomial time [9], [21].

In contrast to submodular minimization, submodular maximization problems are typically hard to solve exactly. Consider the classical problem of maximizing a monotone submodular function subject to a cardinality constraint, $\max\{f(S) : |S| \leq k\}$. It is known that this problem admits a $(1-1/e)$-approximation [18] and this is optimal in two different ways: (i) Given only black-box access to $f(S)$, we cannot achieve a better approximation, unless we ask exponentially many value queries [20]. This holds even if we have unlimited computational power. (ii) In special cases where $f(S)$ has a compact representation on the input, it is NP-hard to achieve an approximation better than $1-1/e$ [7]. The reason why the hardness threshold is the same in both cases is apparently not well understood.

The optimal $(1-1/e)$-approximation for the problem $\max\{f(S) : |S| \leq k\}$ is achieved by a simple greedy algorithm [18], but this seems to be rather coincidental. For other variants of submodular maximization, such as unconstrained (non-monotone) submodular maximization [8], monotone submodular maximization subject to a matroid constraint [19], [3], [24], or submodular maximization subject to linear constraints [13], [14], greedy algorithms achieve suboptimal results. A tool which has proven useful in approaching these problems is *multilinear relaxation*.

Multilinear relaxation. Let us consider a discrete optimization problem $\max\{f(S) : S \in \mathcal{F}\}$, where $f : 2^X \to \mathbb{R}$ is the objective function and $\mathcal{F} \subset 2^X$ is the collection of feasible

*This work was done while the author was at Princeton University. The paper's length is due to merging two submissions on this topic.

solutions. In case f is a linear function, $f(S) = \sum_{j \in S} w_j$, it is natural to replace this problem by a linear programming problem. For a general set function $f(S)$, however, a linear relaxation is not readily available. Instead, the following relaxation has been proposed [3], [24]: For $x \in [0,1]^X$, let $\hat{x}$ denote a random vector in $\{0,1\}^X$ where each coordinate of x_i is rounded independently with expectation x_i. We define

$$F(x) = \mathbf{E}[f(\hat{x})] = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j).$$

This is a *multilinear function* which coincides with $f(S)$ on $\{0,1\}$-vectors. We remark that although we cannot compute the exact value of $F(x)$ for a given $x \in [0,1]^X$ (which would require querying all possible values of $f(S)$), we can compute $F(x)$ approximately, by random sampling. Sometimes this causes technical issues, which we also deal with in this paper.

Instead of the discrete problem $\max\{f(S) : S \in \mathcal{F}\}$, we consider a continuous optimization problem $\max\{F(x) : x \in P(\mathcal{F})\}$, where $P(\mathcal{F})$ is the convex hull of indicator vectors corresponding to $\mathcal{F}$,

$$P(\mathcal{F}) = \{\sum_{S \in \mathcal{F}} \alpha_S \mathbf{1}_S : \sum_{S \in \mathcal{F}} \alpha_S = 1, \alpha_S \geq 0\}.$$

The main reason why the extension $F(x) = \mathbf{E}[f(\hat{x})]$ is useful for submodular maximization problems is that fractional solutions can be often rounded to discrete ones without losing *anything* in terms of the objective value. Then, our ability to solve the multilinear relaxation approximately translates directly into an approximation algorithm for the original problem. In particular, this is true when the collection of feasible solutions forms a matroid.

Pipage rounding was originally developed by Ageev and Sviridenko for rounding solutions in the bipartite matching polytope [2]. The technique was adapted to matroid polytopes by Calinescu et al. [3], who proved that for any submodular function $f(S)$ and any x in the matroid base polytope $B(\mathcal{M})$, the fractional solution x can be rounded to a base $B \in \mathcal{B}$ such that $f(B) \geq F(x)$. This approach leads to an optimal $(1 - 1/e)$-approximation for the Submodular Welfare Problem, and more generally for monotone submodular maximization subject to a matroid constraint [3], [24]. It is also known that the factor of $1-1/e$ is optimal for the Submodular Welfare Problem both in the NP framework [12] and in the value oracle model [17]. Under the assumption that the submodular function $f(S)$ has *curvature* c, there is a $\frac{1}{c}(1 - e^{-c})$-approximation and this is also optimal in the value oracle model [25]. The framework of pipage rounding can be also extended to nonmonotone submodular functions; this presents some additional issues which we discuss in this paper.

For the problem of unconstrained (non-monotone) submodular maximization, a $2/5$-approximation was developed in [8]. This algorithm implicitly uses the multilinear relaxation $\max\{F(x) : x \in [0,1]^X\}$. For *symmetric* submodular functions, it is shown in [8] that a uniformly random solution $x = (1/2, \dots, 1/2)$ gives $F(x) \geq \frac{1}{2} OPT$, and there is no better approximation algorithm in the value oracle model.

Using additional techniques, the multilinear relaxation can be also applied to submodular maximization with knapsack-type constraints ($\sum_{j \in S} c_{ij} \leq 1$). For the problem of maximizing a monotone submodular function subject to a constant number of knapsack constraints, there is a $(1-1/e-\epsilon)$-approximation algorithm for any $\epsilon > 0$ [13]. For maximizing a non-monotone submodular function subject to a constant number of knapsack constraints, a $(1/5 - \epsilon)$-approximation was designed in [14].

One should mention that not all the best known results for submodular maximization have been achieved using the multilinear relaxation. The greedy algorithm yields a $1/(k+1)$-approximation for monotone submodular maximization subject to k matroid constraints [19]. Local search methods have been used to improve this to a $1/(k+\epsilon)$-approximation, and to obtain a $1/(k+1+1/(k-1)+\epsilon)$-approximation for the same problem with a non-monotone submodular function, for any $\epsilon > 0$ [14], [15]. For the problem of maximizing a non-monotone submodular function over the *bases* of a given matroid, local search yields a $(1/6-\epsilon)$-approximation, assuming that the matroid contains two disjoint bases [14].

1.1. Our results

Our main result (Theorem 1.6) is a general hardness construction which produces an inapproximability result in the value oracle model in an automated way, based on what we call the *symmetry gap* for some fixed instance. In this generic fashion, we are able to replicate a number of previously known hardness results (such as the optimality of the factors $1 - 1/e$ and $1/2$ mentioned above), and we also produce new hardness results using this construction (Theorems 1.1, 1.3). Our construction helps explain the particular hardness thresholds obtained under various constraints, by exhibiting a small instance where the threshold can be seen as the gap between the optimal solution and the best symmetric solution. The query complexity results in [8], [17], [25] can be seen in hindsight as special cases of Theorem 1.6, but the construction in this paper is somewhat different and technically more involved than the previous proofs for particular cases.

Before we proceed to describe our general hardness result, we present its implications for two more concrete problems. We also provide closely matching approximation algorithms for these two problems, based on multilinear relaxation. First, we discuss the problem of maximizing a (non-monotone) submodular function subject to a matroid independence constraint. In the following, we assume that the function is given by a value oracle and the matroid is given by a membership oracle.

Theorem 1.1. *There is a randomized* $\frac{1}{4}(-1+\sqrt{5}-o(1)) \doteq 0.309$*-approximation for the problem* $\max\{f(S) : S \in \mathcal{I}\}$*, where* $f(S)$ *is a nonnegative submodular function, and* $\mathcal{I}$ *is a collection of independent sets in a matroid.*

For any $\epsilon > 0$*, a* $(\frac{1}{2}+\epsilon)$*-approximation for this problem (e.g. when* $\mathcal{I} = \{I : |I| \leq \frac{n}{2}\}$*) would require exponentially many value queries.*

Our algorithmic result improves a previously known $(\frac{1}{4} - o(1))$-approximation [14]. The hardness threshold follows from our general result, but also quite easily from [8].

Secondly, we consider the problem of maximizing a (non-monotone) submodular function subject to a matroid *base* constraint. (This generalizes for example the maximum bisection problem in graphs.) We show that the approximability of this problem is related to base packings in the matroid. We use the following definition.

Definition 1.2. *For a matroid* $\mathcal{M}$ *with a collection of bases* $\mathcal{B}$*, the fractional base packing number is the maximum possible value of* $\sum_{B\in\mathcal{B}} \alpha_B$ *for* $\alpha_B \geq 0$ *such that* $\sum_{B\in\mathcal{B}:j\in B} \alpha_B \leq 1$ *for every element* j*.*

Theorem 1.3. *For any* $\nu \in (1,2]$*, there is a randomized* $\frac{1}{2}(1-\frac{1}{\nu}-o(1))$*-approximation for the problem* $\max\{f(S) : S \in \mathcal{B}\}$*, where* $f(S)$ *is a nonnegative submodular function, and* $\mathcal{B}$ *is a collection of bases in a matroid with fractional packing number at least* ν*.*

On the other hand, for any ν *in the form* $\frac{k}{k-1}, k \geq 2$*, and any fixed* $\epsilon > 0$*, a* $(1-\frac{1}{\nu}+\epsilon)$*-approximation for the same problem would require exponentially many value queries.*

In case the matroid contains two disjoint bases ($\nu = 2$), we obtain a $(\frac{1}{4} - o(1)$-approximation, improving the previously known factor of $\frac{1}{6} - o(1)$ [14]. In the range of $\nu \in (1,2]$, our positive and negative results are within a factor of 2. For maximizing a submodular function over the bases of a general matroid, our result implies that there is no constant-factor approximation. This is also a new result.

In the following, we explain the notion of a symmetry gap and our general hardness result.

1.2. The symmetry gap

Consider an instance $\max\{f(S) : S \in \mathcal{F}\}$ which exhibits a certain degree of symmetry. This is formalized by the notion of a *symmetry group* $\mathcal{G}$. We consider permutations $\sigma \in \mathbf{S}(X)$ where $\mathbf{S}(X)$ is the symmetric group of on X; we also use σ for the naturally induced mapping of subsets of X. We say that the instance is invariant under $\mathcal{G} \subset \mathbf{S}(X)$, if for any $\sigma \in \mathcal{G}$ and any set S, $f(S) = f(\sigma(S))$ and $S \in \mathcal{F} \Leftrightarrow \sigma(S) \in \mathcal{F}$. We emphasize that even though we apply σ to sets, it must be derived from a permutation on X. For $x \in [0,1]^X$, we define the "symmetrization of x" as

- $\bar{x} = \mathbf{E}_{\sigma\in\mathcal{G}}[\sigma(x)]$,

where $\sigma(x)$ denotes x with coordinates permuted by σ. Then, we define the symmetry gap as the ratio between the optimal solution of $\max\{F(x) : x \in P(\mathcal{F})\}$ and the best *symmetric* solution of this problem.

Definition 1.4 (Symmetry gap)**.** *Let* $\max\{f(S) : S \in \mathcal{F}\}$ *be an instance on a ground set* X*, which is invariant under* $\mathcal{G} \subset \mathbf{S}(X)$*. Let* $F(x) = \mathbf{E}[f(\hat{x})]$ *be the multilinear extension of* $f(S)$ *and* $P(\mathcal{F}) = conv(\{\mathbf{1}_I : I \in \mathcal{F}\})$ *the polytope associated with* $\mathcal{F}$*. The symmetry gap of* $\max\{f(S) : S \in \mathcal{F}\}$ *is defined as* $\gamma = \overline{OPT}/OPT$ *where*

- $OPT = \max\{F(x) : x \in P(\mathcal{F})\}$.
- $\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{F})\}$.

Next, we need to define the notion of a *refinement* of an instance. The following definition may appear technical, but we believe that this is a natural way to "blow up" an instance. In particular, this operation preserves the types of constraints that we care about, such as cardinality constraints, matroid independence, knapsack constraints, and matroid base constraints.

Definition 1.5 (Refinement)**.** *Let* $\mathcal{F} \subseteq 2^X$*,* $|X| = k$ *and* $|N| = n$*. We say that* $\tilde{\mathcal{F}} \subseteq 2^{N\times X}$ *is a refinement of* $\mathcal{F}$*, if*

- $S \in \tilde{\mathcal{F}}$ *if and only if* $(x_1,\ldots,x_k) \in P(\mathcal{F})$*, where* $x_j = \frac{1}{n}|S \cap (N \times \{j\})|$*.*

Our main result is that the symmetry gap translates automatically into hardness of approximation for refined instances. We emphasize that this is a query-complexity lower bound, and hence independent of assumptions such as $P \neq NP$.

Theorem 1.6. *Let* $\max\{f(S) : S \in \mathcal{F}\}$ *be an instance of nonnegative (optionally monotone) submodular maximization with symmetry gap* $\gamma = \overline{OPT}/OPT$*. Let* $\mathcal{C}$ *be the class of instances* $\max\{\tilde{f}(S) : S \in \tilde{\mathcal{F}}\}$ *where* $\tilde{f}$ *is nonnegative (optionally monotone) submodular and* $\tilde{\mathcal{F}}$ *is a refinement of* $\mathcal{F}$*. Then for every* $\epsilon > 0$*, any* $(1+\epsilon)\gamma$*-approximation algorithm for the class* $\mathcal{C}$ *would require exponentially many value queries to* $\tilde{f}(S)$*.*

We remark that the result holds even if the class $\mathcal{C}$ is restricted to instances which are themselves symmetric under a group similar to $\mathcal{G}$. Moreover, for symmetric instances the algorithmic problem becomes easier and we obtain optimal approximation factors, up to lower-order terms. (See Section 5 for more details.)

The rest of the paper is organized as follows. In Section 2, we show applications of our main hardness result (Theorem 1.6) to concrete cases, in particular how it implies the hardness statements in Theorem 1.1 and 1.3. In Section 3, we present the proof of Theorem 1.6. In Section 4, we prove the algorithmic results in Theorem 1.1 and 1.3. In Section 5, we discuss the special case of symmetric instances. In the Appendix, we present a few basic facts concerning submodular functions and their extensions, our generalization of pipage rounding to nonmonotone submodular functions, and a few other technicalities which would hinder the main exposition.

2. FROM SYMMETRY TO INAPPROXIMABILITY: APPLICATIONS

Before we plunge into the proof of Theorem 1.6, let us show how it can be applied to a number of specific problems. Some of these are hardness results that were proved previously by an ad-hoc method. The last application is a new one (Theorem 1.3).

Nonmonotone submodular maximization. Let $X = \{1, 2\}$ and for any $S \subseteq X$, $f(S) = 1$ if $|S| = 1$, and 0 otherwise. Consider the instance $\max\{f(S) : S \subseteq X\}$. In other words, this is the Max Cut problem on the graph K_2. This instance exhibits a simple symmetry, the group of all (two) permutations on $\{1, 2\}$. We get $OPT = F(1, 0) = F(0, 1) = 1$, while $\overline{OPT} = F(1/2, 1/2) = 1/2$. Hence, the symmetry gap is $1/2$.

Since $f(S)$ is nonegative submodular and there is no constraint on $S \subseteq X$, this will be the case for any refinement of the instance as well. Theorem 1.6 implies immediately the following: any algorithm achieving a better than $1/2$-approximation for nonnegative nonmonotone submodular maximization requires exponentially many value queries (previously known [8]).

Note that the same symmetry gap holds even if we impose some simple constraints: the problems $\max\{f(S) : |S| \leq 1\}$ and $\max\{f(S) : |S| = 1\}$ have the same symmetry gap as above. Hence, the hardness threshold of $1/2$ also holds for nonmonotone submodular maximization under cardinality constraints of the type $|S| \leq n/2$, or $|S| = n/2$. This proves the hardness part of Theorem 1.1. This can be derived quite easily from the construction of [8] as well.

Monotone submodular maximization. Let $X = [k]$ and $f(S) = \min\{|S|, 1\}$. Consider the instance $\max\{f(S) : |S| \leq 1\}$. This instance exhibits the symmetry of all permutations on $[k]$. We get $OPT = F(1, 0, \ldots, 0) = 1$, while $\overline{OPT} = F(1/k, 1/k, \ldots, 1/k) = 1 - (1 - 1/k)^k$.

Here, $f(S)$ is monotone submodular and any refinement of $\mathcal{F}$ is a set system of the type $\tilde{\mathcal{F}} = \{S : |S| \leq \ell\}$. Based on our theorem, this implies that any approximation better than $1 - (1 - 1/k)^k$ for monotone submodular maximization subject to a cardinality constraint would require exponentially many value queries. Since this holds for any fixed k, we get the same hardness hardness result for any $\beta > \lim_{k\to\infty}(1 - (1 - 1/k)^k) = 1 - 1/e$ (previously known [20]).

Submodular welfare. Let $X = [k] \times [k]$, $\mathcal{F} = \{S : S$ contains at most 1 pair (i, j) for each $j\}$, and $f(S) = |\{i : \exists(i, j) \in S\}|$. Consider the instance $\max\{f(S) : S \in \mathcal{F}\}$. This instance can be interpreted as an allocation problem of k items to k players. A set S represents an assignment in the sense that $(i, j) \in S$ if item j is allocated to player i. A player is satisfied is she receives at least 1 item; the objective function is the number of satisfied players.

This instance exhibits the symmetry of all permutations on the items. An optimum solution allocates each item to a different player, and $OPT = k$. The symmetrized optimum is $\overline{OPT} = F(1/k, 1/k, \ldots, 1/k) = k(1 - (1 - 1/k)^k)$. A refinement of this instance can be interpreted as an allocation problem where we have n copies of each item, we still have k players, and the utility functions are monotone submodular. Our theorem implies that for Submodular Welfare with k players, a better approximation factor than $1 - (1 - 1/k)^k$ is impossible.[1]

Submodular maximization over matroid bases. Let $X = A \cup B$, $A = \{a_1, \ldots, a_k\}$, $B = \{b_1, \ldots, b_k\}$ and $\mathcal{F} = \{S : |S \cap A| = 1 \ \& \ |S \cap B| = k - 1\}$. We define $f(S) = \sum_{i=1}^{k} f_i(S)$ where $f_i(S) = 1$ if $a_i \in S$ & $b_i \notin S$, and 0 otherwise. This instance can be viewed as a Max Di-Cut problem on a graph of k disjoint arcs, under the constraint that exactly one arc tail and $k - 1$ arc heads should be on the left-hand side (S). An optimal solution is for example $S = \{a_1, b_2, b_3, \ldots, b_k\}$, which gives $OPT = 1$.

The symmetry here is that we can apply the same permutation to A and B simultaneously. This yields a unique symmetrized solution $\bar{x} = (\frac{1}{k}, \ldots, \frac{1}{k}, 1 - \frac{1}{k}, \ldots, 1 - \frac{1}{k})$, and $\overline{OPT} = F(\bar{x}) = \mathbf{E}[f(\hat{\bar{x}})] = \sum_{i=1}^{k} \mathbf{E}[f_i(\hat{\bar{x}})] = \frac{1}{k}$ (we get each arc in the directed cut with probability $\frac{1}{k^2}$, hence $\mathbf{E}[f_i(\hat{\bar{x}})] = \frac{1}{k}$).

The refined instances are instances of (nonmonotone) submodular maximization over the bases of a matroid, where the ground set is partitioned into $A \cup B$ and we should take a $\frac{1}{k}$-fraction of A and a $(1 - \frac{1}{k})$-fraction of B. (This means that the fractional packing number of bases is $\nu = k/(k - 1)$.) Our theorem implies that for this class of instances, an approximation better than $1/k$ is impossible - this proves the hardness part of Theorem 1.3.

3. FROM SYMMETRY TO INAPPROXIMABILITY: PROOF

On a high level, our proof resembles the constructions of [8], [17]. We construct instances based on continuous functions $F(x), G(x)$, whose optima differ by a gap $(1+\epsilon)\gamma$ for which we want to prove hardness. Then we show that after a certain perturbation, the two types of instances are very hard to distinguish. This paper generalizes the ideas of [8], [17] and brings two new ingredients. First, we show that the functions $F(x), G(x)$, which are "pulled out of the hat" in [8], [17], can be produced in a natural way from the multilinear relaxation of the respective problem, using the notion of a *symmetry gap*. Secondly, the way we perturb these functions is quite delicate and forms the main technical part of the proof. In [8], this step is quite simple. In [17], the perturbation is more complicated, but still relies on properties of the specific functions $F(x), G(x)$ which do not hold in general. The construction that we present

[1] We note that formally, our theorem assumes an oracle model where only the total value can be queried for a given allocation. This is actually enough for the $(1 - 1/e)$-approximation of [24] to work. The hardness result holds even if each player's utility function can be queried separately; this result was proved in [17].

here (Lemma 3.2) uses the symmetry properties of a fixed instance in a generic fashion, and technically it is more involved than the ones in [8], [17]. First, let us present an outline of our construction.

Given an instance $\max\{f(S) : S \in \mathcal{F}\}$ exhibiting a symmetry gap γ, we consider two smooth submodular[2] functions, $F(x)$ and $G(x)$. The first one is the multilinear extension $F(x) = \mathbf{E}[f(\hat{x})]$, while the second one is its symmetrized version $G(x) = F(\bar{x})$. We modify these functions slightly so that we obtain functions $\hat{F}(x)$ and $\hat{G}(x)$ with the following property: For any vector x which is close to its symmetrized version $\bar{x}$, $\hat{F}(x) = \hat{G}(x)$. The functions $\hat{F}(x), \hat{G}(x)$ induce instances of submodular maximization on the refined ground sets. The way we define discrete instances based on $\hat{F}(x), \hat{G}(x)$ is very natural, using the following lemma which has been used in [17], [14].

Lemma 3.1. *Let $F : [0,1]^X \to \mathbb{R}$ be a function with continuous first partial derivatives everywhere, and second partial derivatives almost everywhere. Let $N = [n]$, $n \geq 1$, and define $f : N \times X \to \mathbb{R}$ so that $f(S) = F(x)$ where $x_i = \frac{1}{n}|S \cap (N \times \{i\})|$. Then*

- *If $\frac{\partial F}{\partial x_i} \geq 0$ everywhere for each i, then $f(S)$ is monotone.*
- *If $\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$ almost everywhere for all i, j, then $f(S)$ is submodular.*

The way we construct $\hat{F}(x), \hat{G}(x)$ is such that, given a large enough refinement of the ground set, it is impossible to distinguish the instances corresponding to $\hat{F}(x)$ and $\hat{G}(x)$. As we argue more precisely later, this holds because under an unknown labeling of the ground set, all queries with high probability fall in the region where $\hat{F}(x) = \hat{G}(x)$. The following lemma gives the precise properties of $\hat{F}(x)$ and $\hat{G}(x)$ that we need.

Lemma 3.2. *Consider an instance $\max\{f(S) : S \in \mathcal{F}\}$ invariant under a group of permutations $\mathcal{G}$ on a ground set X. Assume $f : 2^X \to [0, M]$ and $F(x) = \mathbf{E}[f(\hat{x})]$. Let $\bar{x} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$, $OPT = \max\{F(x) : x \in P(\mathcal{F})\}$ and $\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{F})\}$. Fix any $\epsilon > 0$. Then there is $\delta > 0$ and functions $\hat{F}, \hat{G} : [0,1]^X \to \mathbb{R}$ (which are also symmetric with respect to $\mathcal{G}$), satisfying:*

- *Whenever $||x - \bar{x}||^2 \leq \delta$, $\hat{F}(x) = \hat{G}(x)$ and the value depends only on $\bar{x}$.*
- $\max\{\hat{F}(x) : x \in P(\mathcal{F})\} \geq OPT$.
- $\max\{\hat{G}(x) : x \in P(\mathcal{F})\} \leq (1+\epsilon)\overline{OPT}$.
- *If $f(S)$ is monotone, we have $\frac{\partial \hat{F}}{\partial x_i} \geq 0$, $\frac{\partial \hat{G}}{\partial x_i} \geq 0$.*
- *If $f(S)$ is submodular, $\frac{\partial^2 \hat{F}}{\partial x_i \partial x_j} \leq 0$, $\frac{\partial^2 \hat{G}}{\partial x_i \partial x_j} \leq 0$.*

The proof of this lemma is the main technical part of this paper and we defer it to the end of this section. Assuming this lemma, we first finish the proof of the main theorem.

[2]"Smooth submodularity" means the condition $\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$ for all i, j.

Proof of Theorem 1.6: Fix an $\epsilon > 0$. Given a symmetric instance $\max\{f(S) : S \in \mathcal{F}\}$ invariant under $\mathcal{G}$, let $\hat{F}, \hat{G} : [0,1]^X \to \mathbb{R}$ be the two functions provided by Lemma 3.2. We choose a large number n and consider a refinement $\tilde{\mathcal{F}}$ on the ground set $N \times X$, where $N = [n]$. We want to define discrete instances of submodular maximization $\max\{f(S) : S \in \tilde{\mathcal{F}}\}$, $\max\{g(S) : S \in \tilde{\mathcal{F}}\}$ based on the functions $\hat{F}(x), \hat{G}(x)$. However, to confuse the algorithm, we randomize the correspondence of the ground set $N \times X$ to the continuous space $[0,1]^X$ in the following fashion. For each $i \in N$, we choose independently a random permutation $\sigma^{(i)} \in \mathcal{G}$. These permutations are chosen randomly but fixed once the input is presented to an algorithm. This can be viewed as a random shuffle of the labeling of the ground set before we present it to an algorithm.

For every set $S \subseteq N \times X$, we define a vector $\xi(S) \in [0,1]^X$ by

$$\xi_j(S) = \frac{1}{n}\left|\{i \in N : (i, \sigma^{(i)}(j)) \in S\}\right|.$$

In other words, $\xi_j(S)$ measures the fraction of copies of element j contained in S; however, for each i the i-copies of all elements are shuffled by $\sigma^{(i)}$. Then, we define

$$f(S) = \hat{F}(\xi(S)), \qquad g(S) = \hat{G}(\xi(S)).$$

We claim that $f(S)$ and $g(S)$ are submodular (for any fixed ξ). Note that the effect of $\sigma^{(i)}$ is just a renaming (or reshuffling) of the elements of $N \times X$, and hence for the purpose of submodularity we can assume that $\sigma^{(i)} = Id$ for all i. Then, $\xi_j(S) = \frac{1}{n}|S \cap (N \times \{j\})|$. Due to Lemma 3.1, the property $\frac{\partial^2 \hat{F}}{\partial x_i \partial x_j} \leq 0$ implies that $f(S)$ is submodular. In addition, if the original instance was monotone, then $\frac{\partial \hat{F}}{\partial x_j} \geq 0$ and $f(S)$ is monotone as well. The same holds for $g(S)$.

The feasible sets in the refined instance $S \in \tilde{\mathcal{F}}$ are such that the respective vector $\xi(S)$ is in the polytope $P(\mathcal{F})$. The value of $g(S)$ for any feasible solution $S \in \tilde{\mathcal{F}}$ is bounded by $g(S) = \hat{G}(\xi(S)) \leq (1+\epsilon)\overline{OPT}$. On the other hand, let x^* denote a point where the optimum of the continuous problem $\max\{\hat{F}(x) : x \in P(\mathcal{F})\}$ is attained, i.e. $F(x^*) = OPT$. For a large enough n, we can approximate the optimal point x^* arbitrarily well by a rational vector with n in the denominator, which corresponds to a discrete solution $S^* \in \tilde{\mathcal{F}}$ whose value $f(S)$ is arbitrarily close to OPT. Hence, the gap between the optima of the *discrete* optimization problems $\max\{f(S) : S \in \tilde{\mathcal{F}}\}$ and $\max\{g(S) : S \in \tilde{\mathcal{F}}\}$ can be made arbitrarily close to $(1+\epsilon)\overline{OPT}/OPT = (1+\epsilon)\gamma$.

We claim that no algorithm (even randomized) can distinguish between the two instances, $\max\{f(S) : S \in \mathcal{F}'\}$ and $\max\{g(S) : S \in \mathcal{F}'\}$. The way we argue about this is that given our probability distribution over the labelings of the ground set, we show that every deterministic algorithm returns the same solution (of the same value) on both instances with high probability. By Yao's principle, this means that any randomized algorithm also returns the same answer with high probability for some labeling of the input. If an algorithm cannot distinguish between the two instances, it means that it cannot approximate the optimum value within a factor of $(1+\epsilon)\gamma$, for any fixed $\epsilon > 0$.

The key observation is that for any query of the algorithm Q, which is blind to the underlying randomness in $\sigma^{(i)}$, the associated vector $q = \xi(Q)$ is very likely to be close to its symmetrized version $\bar{q}$. To see this, consider a query Q. The associated vector $q = \xi(Q)$ is determined by

$$q_j = \frac{1}{n}|\{i \in N : (i, \sigma^{(i)}(j)) \in Q\}| = \frac{1}{n}\sum_{i=1}^{n} Q_{ij}$$

where Q_{ij} is the indicator variable of the event $(i, \sigma^{(i)}(j)) \in Q$. This is a random event due to the randomness in $\sigma^{(i)}$. We have

$$\mathbf{E}[Q_{ij}] = \Pr[Q_{ij} = 1] = \Pr_{\sigma^{(i)} \in \mathcal{G}}[(i, \sigma^{(i)}(j)) \in Q].$$

Adding up these expectations over $i \in N$, we get

$$\sum_{i=1}^{n} \mathbf{E}[Q_{ij}] = \sum_{i=1}^{n} \Pr_{\sigma^{(i)} \in \mathcal{G}}[(i, \sigma^{(i)}(j)) \in Q]$$

$$= \mathbf{E}_{\sigma \in \mathcal{G}}[|\{i \in N : (i, \sigma(j)) \in Q\}|].$$

For the purposes of expectation, the independence of $\sigma^{(1)}, \ldots, \sigma^{(n)}$ is irrelevant and that is why we can drop the indices. On the other hand, consider the symmetrized vector $\bar{q}$:

$$\bar{q}_j = \mathbf{E}_{\sigma \in \mathcal{G}}[q_{\sigma(j)}] = \frac{1}{n}\mathbf{E}_{\sigma \in \mathcal{G}}[|\{i \in N : (i, \sigma^{(i)}(\sigma(j))) \in Q\}|]$$

$$= \frac{1}{n}\mathbf{E}_{\sigma \in \mathcal{G}}[|\{i \in N : (i, \sigma(j)) \in Q\}|]$$

using the fact that the distribution of $\sigma^{(i)} \circ \sigma$ is the same as the distribution of σ - uniformly random over $\mathcal{G}$. Note that the vector q depends on the random permutations $\sigma^{(i)}$ but the symmetrized vector $\bar{q}$ does not; this will be also useful in the following. For now, we summarize that

$$\bar{q}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}[Q_{ij}].$$

Since each permutation $\sigma^{(i)}$ is chosen independently, the random variables $\{Q_{ij} : 1 \leq i \leq n\}$ are independent (for a fixed j). We can apply Chernoff's bound (see e.g. [1], Corollary A.1.7):

$$\Pr\left[\left|\sum_{i=1}^{n} Q_{ij} - \sum_{i=1}^{n} \mathbf{E}[Q_{ij}]\right| > a\right] < 2e^{-2a^2/n}.$$

Using $q_j = \frac{1}{n}\sum_{i=1}^{n} Q_{ij}$, $\bar{q}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}[Q_{ij}]$ and setting $a = n\sqrt{\delta/|X|}$, we obtain

$$\Pr\left[|q_j - \bar{q}_j| > \sqrt{\frac{\delta}{|X|}}\right] < 2e^{-2n\delta/|X|}.$$

By the union bound,

$$\Pr[||q - \bar{q}||^2 > \delta] \leq \sum_{j \in X} \Pr[|q_j - \bar{q}_j|^2 > \frac{\delta}{|X|}] < 2|X|e^{-2n\delta/|X|}.$$

Note that while δ and $|X|$ are constants, n grows as the size of the refinement and hence the probability is exponentially small in the size of the ground set $N \times X$.

As long as $D(q) = ||q - \bar{q}||^2 \leq \delta$ for every query issued by the algorithm, the answers do not depend on the randomness of the input. This is because then the values of $\hat{F}(q)$ and $\hat{G}(q)$ depend only on $\bar{q}$, which is independent of the random permutations $\sigma^{(i)}$, as we argued above. Therefore, assuming that $D(q) \leq \delta$ for each query, the algorithm will always follow the same path of computation and issue the same sequence of queries $\mathcal{S}$. (Note that this is just a fixed sequence which can be written down before we started running the algorithm.) Assume that $|\mathcal{S}|$ is subexponential in n. It happens with high probability that $D(q) \leq \delta$ for all $Q \in \mathcal{S}$. Then, the algorithm indeed follows this path of computation and gives the same answer. The answer does not depend on whether the objective function is $f(S)$ or $g(S)$. ■

We remark that since $\hat{F}(x)$ and $\hat{G}(x)$ are symmetric under $\mathcal{G}$, the refined instances that we define are invariant with respect to the following symmetries: permute the copies of each element in an arbitrary way, and permute the classes of copies according to any permutation $\sigma \in \mathcal{G}$. This means that the hardness result also holds for instances satisfying such symmetry properties.

It remains to prove Lemma 3.2. Before we move to the final construction of $\hat{F}(x)$ and $\hat{G}(x)$, we construct as an intermediate step a function $\tilde{F}(x)$ which is helpful in the analysis.

Construction: Let us construct a function $\tilde{F}(x)$ which satisfies (roughly) the following:

- For x "sufficiently close" to $\bar{x}$, $\tilde{F}(x) = G(x)$.
- For x "sufficiently far away" from $\bar{x}$, $\tilde{F}(x) \simeq F(x)$.
- The function $\tilde{F}(x)$ is not exactly smooth submodular, but the conditions are not violated too badly.

Once we have $\tilde{F}(x)$, we can fix it to obtain a smooth submodular function $\hat{F}(x)$, which is still close to the original function $F(x)$. We also fix $G(x)$ in the same way, to obtain a function $\hat{G}(x)$ which is equal to $\hat{F}(x)$ whenever x is close to $\bar{x}$. We defer this step until the end.

We define $\tilde{F}(x)$ as a convex linear combination of $F(x)$ and $G(x)$, guided by a "smooth transition" function, depending on the distance of x from $\bar{x}$. The form that we use[3] is as follows:

$$\tilde{F}(x) = (1 - \phi(D(x)))F(x) - \phi(D(x))G(x)$$

where $\phi : \mathbb{R}_+ \to [0,1]$ is a suitable smooth function, and

$$D(x) = ||x - \bar{x}||^2 = \sum_i (x_i - \bar{x}_i)^2.$$

The idea is that when x is close to $\bar{x}$, the convex linear combination should give most of the weight to $G(x)$, and the weight shifts gradually to $F(x)$ as x gets further away from $\bar{x}$. Therefore, $\phi(t) = 1$ in a small interval $t \in [0, \delta]$, and $\phi(t)$ tends to 0 as t increases. This guarantees that $\tilde{F}(x) = G(x)$ whenever $D(x) = ||x - \bar{x}||^2 \leq \delta$. We defer the precise construction of $\phi(t)$ to Lemma 3.6, after we determine what properties we need from $\phi(t)$. Note that regardless of the definition of $\phi(t)$, $\tilde{F}(x)$ is also symmetric with respect to $\mathcal{G}$, since $F(x), G(x)$ and $D(x)$ are.

Analysis of the construction: Due to the construction of $\tilde{F}(x)$, it is clear that when $D(x) = ||x - \bar{x}||^2$ is small, $\tilde{F}(\bar{x}) = G(\bar{x})$, whereas when $D(x)$ is large, $\tilde{F}(x) \simeq F(x)$. The main issue, however, is whether we can say something about the first and second partial derivatives of $\tilde{F}(x)$. This is crucial for the properties of monotonicity and submodularity, which we would like to preserve. Let us write $\tilde{F}(x)$ as

$$\tilde{F}(x) = F(x) - \phi(D(x))H(x)$$

where $H(x) = F(x) - G(x)$. By differentiating once, we get

$$\frac{\partial \tilde{F}}{\partial x_i} = \frac{\partial F}{\partial x_i} - \phi(D(x))\frac{\partial H}{\partial x_i} - \phi'(D(x))\frac{\partial D}{\partial x_i}H(x) \quad (1)$$

and by differentiating twice,

$$\frac{\partial^2 \tilde{F}}{\partial x_j \partial x_i} = \frac{\partial^2 F}{\partial x_i \partial x_j} - \phi(D(x))\frac{\partial^2 H}{\partial x_i \partial x_j} - \phi''(D(x))\frac{\partial D}{\partial x_j}\frac{\partial D}{\partial x_i}H(x)$$

$$-\phi'(D(x))\left(\frac{\partial D}{\partial x_j}\frac{\partial H}{\partial x_i} + \frac{\partial^2 D}{\partial x_i \partial x_j}H(x) + \frac{\partial D}{\partial x_i}\frac{\partial H}{\partial x_j}\right). \quad (2)$$

The first two terms on the right-hand side of both (1) and (2)are not bothering us, because they form convex linear combinations of the derivatives of $F(x)$ and $G(x)$, which have the properties that we need. The remaining terms might cause problems, however, and we need to estimate them.

Our strategy is to define $\phi(t)$ in such a way that it eliminates the influence of partial derivatives of D and H where they become too large. Roughly speaking, D and H have negligible partial derivatives when x is very close to $\bar{x}$. As x moves away from $\bar{x}$, the partial derivatives grow but then the behavior of $\phi(t)$ must be such that their influence is supressed.

We start with the following important claim. [4]

Lemma 3.3. *Assume that $F : [0,1]^X \to \mathbb{R}$ is invariant under a group of permutations of coordinates $\mathcal{G}$. Let $G(x) = F(\bar{x})$, where $\bar{x} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$. Then for any $x \in [0,1]^X$,*

$$\nabla G|_x = \nabla F|_{\bar{x}}.$$

Proof: To avoid confusion, we use x_i for the arguments of the functions F and G, and u, $\bar{u}$, etc. for points where their partial derivatives are evaluated. To rephrase, we want to prove that for any point u and any coordinate i, $\frac{\partial G}{\partial x_i}\Big|_{x=u} = \frac{\partial F}{\partial x_i}\Big|_{x=\bar{u}}$.

First, consider $F(x)$. We assume that $F(x)$ is invariant under a group of permutations of coordinates $\mathcal{G}$, i.e. $F(x) = F(\sigma(x))$ for any $\sigma \in \mathcal{G}$. Differentiating both sides at $x = u$, we get by the chain rule:

$$\frac{\partial F}{\partial x_i}\Big|_{x=u} = \sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\sigma(u)} \frac{\partial}{\partial x_i}(\sigma(x))_j = \sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\sigma(u)} \frac{\partial x_{\sigma(j)}}{\partial x_i}.$$

Here, $\frac{\partial x_{\sigma(j)}}{\partial x_i} = 1$ if $\sigma(j) = i$, and 0 otherwise. Therefore,

$$\frac{\partial F}{\partial x_i}\Big|_{x=u} = \frac{\partial F}{\partial x_{\sigma^{-1}(i)}}\Big|_{x=\sigma(u)}.$$

Now, if we evaluate the left-hand side at $\bar{u}$, the right-hand side is evaluated at $\sigma(\bar{u}) = \bar{u}$, and hence for any i and any $\sigma \in \mathcal{G}$,

$$\frac{\partial F}{\partial x_i}\Big|_{x=\bar{u}} = \frac{\partial F}{\partial x_{\sigma^{-1}(i)}}\Big|_{x=\bar{u}}. \quad (3)$$

Turning to $G(x) = F(\bar{x})$, let us write $\frac{\partial G}{\partial x_i}$ using the chain rule:

$$\frac{\partial G}{\partial x_i}\Big|_{x=u} = \frac{\partial}{\partial x_i}F(\bar{x})\Big|_{x=u} = \sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\bar{u}} \cdot \frac{\partial \bar{x}_j}{\partial x_i}.$$

We have $\bar{x}_j = \mathbf{E}_{\sigma \in \mathcal{G}}[x_{\sigma(j)}]$, and so

$$\frac{\partial G}{\partial x_i}\Big|_{x=u} = \sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\bar{u}} \cdot \frac{\partial}{\partial x_i}\mathbf{E}_{\sigma \in \mathcal{G}}[x_{\sigma(j)}]$$

$$= \mathbf{E}_{\sigma \in \mathcal{G}}\left[\sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\bar{u}} \cdot \frac{\partial x_{\sigma(j)}}{\partial x_i}\right].$$

Again, $\frac{\partial x_{\sigma(j)}}{\partial x_i} = 1$ if $\sigma(j) = i$ and 0 otherwise. Consequently, we obtain

$$\frac{\partial G}{\partial x_i}\Big|_{x=u} = \mathbf{E}_{\sigma \in \mathcal{G}}\left[\frac{\partial F}{\partial x_{\sigma^{-1}(i)}}\Big|_{x=\bar{u}}\right] = \frac{\partial F}{\partial x_i}\Big|_{x=\bar{u}}$$

where we used Eq. (3) to remove the dependence on $\sigma \in \mathcal{G}$. ■

[3] We remark that a construction analogous to [17] would be $\tilde{F}(x) = F(x) - \phi(H(x))$ where $H(x) = F(x) - G(x)$. While this makes the analysis easier in [17], it cannot be used in general. Roughly speaking, the problem is that in general the partial derivatives of $H(x)$ are not bounded in any way by the value of $H(x)$.

[4] We remind the reader that ∇F, the gradient of F, is a vector whose coordinates are the first partial derivatives $\frac{\partial F}{\partial x_i}$. We denote by $\nabla F|_x$ the gradient evaluated at x.

Observe that the symmetrization operation $\bar{x}$ is idempotent, i.e. $\bar{\bar{x}} = \bar{x}$. Because of this, we also get $\nabla G|_{\bar{x}} = \nabla F|_{\bar{x}}$. Note that $G(\bar{x}) = F(\bar{x})$ follows from the definition, but it is not obvious that the same holds for gradients, since their definition involves points where $G(x) \neq F(x)$. For second partial derivatives, the equality no longer holds, as can be seen from a simple example such as $F(x_1, x_2) = 1 - (1 - x_1)(1 - x_2)$.

Next, we show that the functions $F(x)$ and $G(x)$ are very similar in the close vicinity of the region where $\bar{x} = x$. Recall our definitions: $H(x) = F(x) - G(x)$, $D(x) = ||x - \bar{x}||^2$. Based on Lemma 3.3, we know that $H(\bar{x}) = 0$ and $\nabla H|_{\bar{x}} = 0$. In the following lemmas, we present bounds on $H(x)$, $D(x)$ and their partial derivatives.

Lemma 3.4. *Let $f : 2^X \to [0, M]$ be invariant with respect to a permutation group $\mathcal{G}$. Let $\bar{x} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$, $D(x) = ||x - \bar{x}||^2$ and $H(x) = F(x) - G(x)$ where $F(x) = \mathbf{E}[f(\hat{x})]$ and $G(x) = F(\bar{x})$. Then*

- $|\frac{\partial^2 H}{\partial x_i \partial x_j}| \leq 8M$ *everywhere, for all i, j*
- $||\nabla H(x)|| \leq 8M|X|\sqrt{D(x)}$
- $|H(x)| \leq 8M|X| \cdot D(x)$.

Proof: First, let us get a bound on the second partial derivatives. We have

$$\frac{\partial^2 F}{\partial x_i \partial x_j} = \mathbf{E}[f(\hat{x} \vee (\mathbf{e}_i + \mathbf{e}_j)) - f(\hat{x} \vee \mathbf{e}_i) - f(\hat{x} \vee \mathbf{e}_j) + f(\hat{x})]$$

(see [24]). Consequently,

$$\Big|\frac{\partial^2 F}{\partial x_i \partial x_j}\Big| \leq 4 \max |f(S)| = 4M.$$

It is a little bit more involved to analyze $\frac{\partial^2 G}{\partial x_i \partial x_j}$. Since $G(x) = F(\bar{x})$ and $\bar{x} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$, we get by the chain rule:

$$\frac{\partial^2 G}{\partial x_i \partial x_j} = \sum_{k,\ell} \frac{\partial^2 F}{\partial x_k \partial x_\ell} \frac{\partial \bar{x}_k}{\partial x_i} \frac{\partial \bar{x}_\ell}{\partial x_j}$$

$$= \mathbf{E}_{\sigma,\tau \in \mathcal{G}} \left[\sum_{k,\ell} \frac{\partial^2 F}{\partial x_k \partial x_\ell} \frac{\partial x_{\sigma(k)}}{\partial x_i} \frac{\partial x_{\tau(\ell)}}{\partial x_j} \right].$$

It is useful here to use the Kronecker symbol, $\delta_{i,j}$, which is 1 if $i = j$ and 0 otherwise. Note that $\frac{\partial x_{\sigma(k)}}{\partial x_i} = \delta_{i,\sigma(k)} = \delta_{\sigma^{-1}(i),k}$, etc. Using this notation, we get

$$\frac{\partial^2 G}{\partial x_i \partial x_j} = \mathbf{E}_{\sigma,\tau \in \mathcal{G}} \left[\sum_{k,\ell} \frac{\partial^2 F}{\partial x_k \partial x_\ell} \delta_{\sigma^{-1}(i),k} \delta_{\sigma^{-1}(j),\ell} \right]$$

$$= \mathbf{E}_{\sigma,\tau \in \mathcal{G}} \left[\frac{\partial^2 F}{\partial x_{\sigma^{-1}(i)} \partial x_{\tau^{-1}(j)}} \right],$$

$$\Big|\frac{\partial^2 G}{\partial x_i \partial x_j}\Big| \leq \mathbf{E}_{\sigma,\tau \in \mathcal{G}} \left[\Big| \frac{\partial^2 F}{\partial x_{\sigma^{-1}(i)} \partial x_{\tau^{-1}(j)}} \Big| \right] \leq 4M$$

and

$$\Big|\frac{\partial^2 H}{\partial x_i \partial x_j}\Big| = \Big|\frac{\partial^2 F}{\partial x_i \partial x_j} - \frac{\partial^2 G}{\partial x_i \partial x_j}\Big| \leq 8M.$$

Next, we estimate $\frac{\partial H}{\partial x_i}$ at a given point u, depending on its distance from $\bar{u}$. Consider the line segment between $\bar{u}$ and u. The function $H(x) = F(x) - G(x)$ is C_∞-differentiable, and hence we can apply the mean value theorem to $\frac{\partial H}{\partial x_i}$: There exists a point $\tilde{u}$ on the line segment $[\bar{u}, u]$ such that

$$\frac{\partial H}{\partial x_i}\Big|_{x=u} - \frac{\partial H}{\partial x_i}\Big|_{x=\bar{u}} = \sum_j \frac{\partial^2 H}{\partial x_j \partial x_i}\Big|_{x=\tilde{u}} (u - \bar{u})_j.$$

Recall that $\frac{\partial H}{\partial x_i}\Big|_{x=\bar{u}} = 0$. Applying the Cauchy-Schwartz inequality to the right-hand side, we get

$$\left(\frac{\partial H}{\partial x_i}\Big|_{x=u}\right)^2 \leq \sum_j \left(\frac{\partial^2 H}{\partial x_j \partial x_i}\Big|_{x=\tilde{u}}\right)^2 ||u - \bar{u}||^2$$

$$\leq (8M)^2 |X| ||u - \bar{u}||^2.$$

Adding up over all i, we obtain

$$||\nabla H(u)||^2 \leq \sum_{i,j} \left(\frac{\partial^2 H}{\partial x_j \partial x_i}\Big|_{x=\tilde{u}}\right)^2 ||u - \bar{u}||^2$$

$$\leq (8M|X|)^2 ||u - \bar{u}||^2.$$

Finally, we estimate the growth of $H(u)$. Again, by the mean value theorem, there is a point $\tilde{u}$ on the line segment $[\bar{u}, u]$, such that

$$H(u) - H(\bar{u}) = \sum_i \frac{\partial H}{\partial x_i}\Big|_{x=\tilde{u}} (u - \bar{u})_i.$$

Using $H(\bar{u}) = 0$, the Cauchy-Schwartz inequality and the above bound on ∇H,

$$(H(u))^2 \leq ||\nabla H|_{\tilde{u}}||^2 ||u - \bar{u}||^2 \leq (8M|X|)^2 ||\tilde{u} - \bar{u}||^2 ||u - \bar{u}||^2,$$

$$|H(u)| \leq 8M|X| \cdot ||\tilde{u} - \bar{u}|| \cdot ||u - \bar{u}||.$$

Clearly, $||\tilde{u} - \bar{u}|| \leq ||u - \bar{u}||$, and therefore

$$|H(u)| \leq 8M|X| \cdot ||u - \bar{u}||^2.$$

■

Lemma 3.5. *For the function $D(x) = ||x - \bar{x}||^2$, we have*

- $\nabla D = 2(x - \bar{x})$, *and therefore* $||\nabla D|| = 2\sqrt{D(x)}$.
- *For all i, j,* $|\frac{\partial^2 D}{\partial x_i \partial x_j}| \leq 2$.

Proof: Let us write $D(x)$ as

$$D(x) = \sum_i (x_i - \bar{x})^2 = \sum_i (\mathbf{E}_{\sigma \in \mathcal{G}}[x_i - x_{\sigma(i)}])^2.$$

Taking the first partial derivative,

$$\frac{\partial D}{\partial x_j} = 2 \sum_i \mathbf{E}_{\sigma \in \mathcal{G}}[x_i - x_{\sigma(i)}] \frac{\partial}{\partial x_j} \mathbf{E}_{\tau \in \mathcal{G}}[x_i - x_{\tau(i)}].$$

As before, we have $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$, etc. Using this notation, we get

$$\begin{aligned} \frac{\partial D}{\partial x_j} &= 2\sum_i \mathbf{E}_{\sigma\in\mathcal{G}}[x_i - x_{\sigma(i)}]\mathbf{E}_{\tau\in\mathcal{G}}[\delta_{ij} - \delta_{\tau(i),j}] \\ &= 2\sum_i \mathbf{E}_{\sigma,\tau\in\mathcal{G}}[(x_i - x_{\sigma(i)})(\delta_{ij} - \delta_{i,\tau^{-1}(j)})] \\ &= 2\ \mathbf{E}_{\sigma,\tau\in\mathcal{G}}[x_j - x_{\sigma(j)} - x_{\tau^{-1}(j)} + x_{\sigma(\tau^{-1}(j))}]. \end{aligned}$$

Since the distributions of $\sigma(j)$, $\tau^{-1}(j)$ and $\sigma(\tau^{-1}(j))$ are the same, we obtain

$$\frac{\partial D}{\partial x_j} = 2\ \mathbf{E}_{\sigma\in\mathcal{G}}[x_j - x_{\sigma(j)}] = 2(x_j - \bar{x}_j)$$

and

$$||\nabla D||^2 = \sum_j \left|\frac{\partial D}{\partial x_j}\right|^2 = 4\sum_j (x_j - \bar{x}_j)^2 = 4D(x).$$

Finally, the second partial derivatives are

$$\frac{\partial^2 D}{\partial x_i \partial x_j} = 2\frac{\partial}{\partial x_i}(x_j - \bar{x}_j) = 2(\delta_{ij} - \mathbf{E}_{\sigma\in\mathcal{G}}[\delta_{i,\sigma(j)}])$$

which is clearly bounded by 2 in the absolute value. ■

Now we come back to $\tilde{F}(x)$ and its partial derivatives. Recall equations (1) and (2). The problematic terms are those involving $\phi'(D(x))$ and $\phi''(D(x))$. Using our bounds on $H(x)$, $D(x)$ and their derivatives, however, we notice that $\phi'(D(x))$ always appears with factors on the order of $D(x)$ and $\phi''(D(x))$ appears with factors on the order of $(D(x))^2$. Thus, it is sufficient if $\phi(t)$ is defined so that we have control over $t\phi'(t)$ and $t^2\phi''(t)$. The following lemma describes the function that we need.

Lemma 3.6. *For any $\alpha, \beta > 0$, there is $\delta > 0$ ($\delta << \beta$) and a differentiable function $\phi : \mathbb{R}_+ \to [0,1]$ such that*

- *For $t \le \delta$, $\phi(t) = 1$.*
- *For $t \ge \beta$, $\phi(t) < e^{-1/\alpha}$.*
- *For all $t \ge 0$, $|t\phi'(t)| \le 4\alpha$.*
- *For all $t \ge 0$, $|t^2\phi''(t)| \le 10\alpha$.*

Proof: First, observe that the quantities $t\phi'(t)$ and $t^2\phi''(t)$ are invariant under a scaling of t. Therefore, we can assume without loss of generality that $\beta > 0$ is a value of our choice, for example $\beta = e^{1/(2\alpha^2)} + 1$. If we want to modify the result for a different value of β, we can just scale the argument t and the constant δ by a suitable factor; the conditions still hold.

We can assume that $\alpha \in (0, 1/8)$ because for larger α, the statement only gets weaker. As we argued, we can assume WLOG that $\beta = e^{1/(2\alpha^2)} + 1$. We set $\delta = 1$ and $\delta_2 = 1 + (1+\alpha)^{-1/2} \le 2$. (We remind the reader that these values will be rescaled depending on the actual value of β.) We define the function as follows:

1) $\phi(t) = 1$ for $t \in [0, \delta]$.
2) $\phi(t) = 1 - \alpha(t-1)^2$ for $t \in [\delta, \delta_2]$.
3) $\phi(t) = (1+\alpha)^{-1-\alpha}(t-1)^{-2\alpha}$ for $t \in [\delta_2, \infty)$.

Let's verify the properties of $\phi(t)$. For $t \in [0,\delta]$, we have $\phi'(t) = \phi''(t) = 0$. For $t \in [\delta, \delta_2]$, we have

$$\phi'(t) = -2\alpha\,(t-1)\,, \qquad \phi''(t) = -2\alpha,$$

and for $t \in [\delta_2, \infty)$,

$$\phi'(t) = -\frac{2\alpha}{(1+\alpha)^{1+\alpha}}\,(t-1)^{-2\alpha-1}\,,$$

$$\phi''(t) = \frac{2\alpha(1+2\alpha)}{(1+\alpha)^{1+\alpha}}\,(t-1)^{-2\alpha-2}\,.$$

First, we check that the values and first derivatives agree at the breakpoints. For $t = \delta = 1$, we get $\phi(1) = 1$ and $\phi'(1) = 0$. For $t = \delta_2 = 1 + (1+\alpha)^{-1/2}$, we get $\phi(\delta_2) = (1+\alpha)^{-1}$ and $\phi'(\delta_2) = -2\alpha(1+\alpha)^{-1/2}$. Next, we need to check is that $\phi(t)$ is very small for $t \ge \beta$. The function is decreasing for $t > \beta$, therefore it is enough to check $t = \beta = e^{1/(2\alpha^2)} + 1$:

$$\phi\,(\beta) = (1+\alpha)^{-1-\alpha}(\beta-1)^{-2\alpha} \le (\beta-1)^{-2\alpha} = e^{-1/\alpha}.$$

The derivative bounds are satisfied trivially for $t \in [0,\delta]$. For $t \in [\delta, \delta_2]$, using $t - 1 \le (1+\alpha)^{-1/2}$,

$$|t\phi'(t)| = 2\alpha{\cdot}t\,(t-1) \le 2\alpha(1{+}(1{+}\alpha)^{-1/2})(1{+}\alpha)^{-1/2} \le 4\alpha$$

and

$$|t^2\phi''(t)| = 2\alpha\cdot t^2 \le 2\alpha(1+(1+\alpha)^{-1/2})^2 \le 8\alpha.$$

For $t \in [\delta_2, \infty)$, using $t - 1 \ge (1+\alpha)^{-1/2}$,

$$|t\phi'(t)| = t\cdot\frac{2\alpha}{(1+\alpha)^{1+\alpha}}\,(t-1)^{-2\alpha-1}$$

$$= \frac{2\alpha}{1+\alpha}\left((1+\alpha)\,(t-1)^2\right)^{-\alpha}\frac{t}{t-1} \le \frac{2\alpha}{1+\alpha}\cdot\frac{t}{t-1}$$

$$\le \frac{2\alpha}{1+\alpha}\cdot\frac{1+(1+\alpha)^{-1/2}}{(1+\alpha)^{-1/2}} = 2\alpha\cdot\frac{1+(1+\alpha)^{-1/2}}{(1+\alpha)^{1/2}} \le 4\alpha$$

and

$$|t^2\phi''(t)| = t^2\cdot\frac{2\alpha(1+2\alpha)}{(1+\alpha)^{1+\alpha}}\,(t-1)^{-2\alpha-2}$$

$$= \frac{2\alpha(1+2\alpha)}{1+\alpha}\left((1+\alpha)\,(t-1)^2\right)^{-\alpha}\left(\frac{t}{t-1}\right)^2$$

$$\le \frac{2\alpha(1+2\alpha)}{1+\alpha}\cdot\left(\frac{1+(1+\alpha)^{-1/2}}{(1+\alpha)^{-1/2}}\right)^2 \le 8\alpha(1{+}2\alpha) \le 10\alpha$$

for $\alpha \in (0, 1/8)$. ■

Using the bounds from Lemmas 3.4, 3.5 and 3.6, we can prove bounds on the derivatives of $\tilde{F}(x)$.

Lemma 3.7. *Let $\tilde{F}(x) = (1 - \phi(D(x)))F(x) + \phi(D(x))G(x)$ where $F(x) = \mathbf{E}[f(\hat{x})]$, $f : 2^X \to [0, M]$, $G(x) = F(\bar{x})$, $D(x) = ||x - \bar{x}||^2$ are as above and $\phi(t)$ is* *provided by Lemma 3.6 for a given $\alpha > 0$. Then, assuming $\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$,*

$$\frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} \leq 512M|X|\alpha.$$

If, in addition, $\frac{\partial F}{\partial x_i} \geq 0$, then

$$\frac{\partial \tilde{F}}{\partial x_i} \geq -64M|X|\alpha.$$

Proof: We have $\tilde{F}(x) = F(x) - \phi(D(x))H(x)$ where $H(x) = F(x) - G(x)$. By differentiating once, we get

$$\frac{\partial \tilde{F}}{\partial x_i} = \frac{\partial F}{\partial x_i} - \phi(D(x))\frac{\partial H}{\partial x_i} - \phi'(D(x))\frac{\partial D}{\partial x_i}H(x),$$

i.e.

$$\Big|\frac{\partial \tilde{F}}{\partial x_i} - \left(\frac{\partial F}{\partial x_i} - \phi(D(x))\frac{\partial H}{\partial x_i}\right)\Big| = \Big|\phi'(D(x))\frac{\partial D}{\partial x_i}H(x)\Big|.$$

By Lemma 3.4 and 3.5, we have $|\frac{\partial D}{\partial x_i}| = 2|x - \bar{x}_i| \leq 2$ and $|H(x)| \leq 8M|X| \cdot D(x)$. Therefore,

$$\Big|\frac{\partial \tilde{F}}{\partial x_i} - \left(\frac{\partial F}{\partial x_i} - \phi(D(x))\frac{\partial H}{\partial x_i}\right)\Big| \leq 16M|X|D(x) \cdot \Big|\phi'(D(x))\Big|.$$

By Lemma 3.6, $|D(x)\phi'(D(x))| \leq 4\alpha$, and

$$\Big|\frac{\partial \tilde{F}}{\partial x_i} - \left(\frac{\partial F}{\partial x_i} - \phi(D(x))\frac{\partial H}{\partial x_i}\right)\Big| \leq 64M|X|\alpha.$$

Assuming that $\frac{\partial F}{\partial x_i} \geq 0$, we also have $\frac{\partial G}{\partial x_i} \geq 0$ (see Lemma 3.3) and therefore, $\frac{\partial F}{\partial x_i} - \phi(D(x))\frac{\partial H}{\partial x_i} = (1 - \phi(D(x)))\frac{\partial F}{\partial x_i} + \phi(D(x))\frac{\partial G}{\partial x_i} \geq 0$. Consequently,

$$\frac{\partial \tilde{F}}{\partial x_i} \geq -64M|X|\alpha.$$

By differentiating $\tilde{F}$ twice, we obtain

$$\frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} = \frac{\partial^2 F}{\partial x_i \partial x_j} - \phi(D(x))\frac{\partial^2 H}{\partial x_i \partial x_j} - \phi''(D(x))\frac{\partial D}{\partial x_i}\frac{\partial D}{\partial x_j}H(x)$$
$$-\phi'(D(x))\left(\frac{\partial D}{\partial x_j}\frac{\partial H}{\partial x_i} + \frac{\partial^2 D}{\partial x_i \partial x_j}H(x) + \frac{\partial D}{\partial x_i}\frac{\partial H}{\partial x_j}\right).$$

Again, we use Lemma 3.4 and 3.5 to bound $|H(x)| \leq 8M|X|D(x)$, $|\frac{\partial H}{\partial x_i}| \leq 8M|X|\sqrt{D(x)}$, $|\frac{\partial^2 H}{\partial x_i \partial x_j}| \leq 8M$, $|\frac{\partial D}{\partial x_i}| \leq 2\sqrt{D(x)}$ and $|\frac{\partial^2 D}{\partial x_i \partial x_j}| \leq 2$. We get

$$\Big|\frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} - \left(\frac{\partial^2 F}{\partial x_i \partial x_j} - \phi(D(x))\frac{\partial^2 H}{\partial x_i \partial x_j}\right)\Big|$$
$$\leq 48\Big|\phi'(D(x))\Big|M|X|D(x) + 32\Big|\phi''(D(x))\Big|M|X|(D(x))^2$$

Observe that $\phi'(D(x))$ appears with the first power of $D(x)$ and $\phi''(D(x))$ appears with the second power of $D(x)$. By Lemma 3.6, $|D(x)\phi'(D(x))| \leq 4\alpha$ and $|D(x)^2\phi''(D(x))| \leq 10\alpha$. Therefore,

$$\Big|\frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} - \left(\frac{\partial^2 F}{\partial x_i \partial x_j} - \phi(D(x))\frac{\partial^2 H}{\partial x_i \partial x_j}\right)\Big|$$
$$\leq 192M|X|\alpha + 320M|X|\alpha = 512M|X|\alpha.$$

We assume that $\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$. Then also $\frac{\partial^2 G}{\partial x_i \partial x_j} \leq 0$ (see the proof of Lemma 3.4) and $\frac{\partial^2 F}{\partial x_i \partial x_j} - \phi(D(x))\frac{\partial^2 H}{\partial x_i \partial x_j} = \phi(D(x))\frac{\partial^2 F}{\partial x_i \partial x_j} + (1 - \phi(D(x)))\frac{\partial^2 G}{\partial x_i \partial x_j} \leq 0$. We obtain

$$\frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} \leq 512M|X|\alpha.$$

■

Finally, we can finish the proof of Lemma 3.2.

Proof of Lemma 3.2: Let $f : 2^X \to [0, M]$, $OPT = \max\{F(x) : x \in P(\mathcal{F})\} = F(x^*)$ and $\beta = D(x^*) = ||x^* - \bar{x}^*||^2$. Let $\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{F})\}$. Given $\epsilon > 0$, let $\alpha = \frac{\overline{OPT}}{768M|X|^3}\epsilon$. For these values of $\alpha, \beta > 0$, let $\delta > 0$ and $\phi : \mathbb{R}_+ \to [0, 1]$ be provided by Lemma 3.6. We define

$$\tilde{F}(x) = (1 - \phi(D(x)))F(x) + \phi(D(x))G(x).$$

Lemma 3.7 provides bounds on the first and second partial derivatives of $\tilde{F}(x)$. Finally, we have to modify $\tilde{F}(x)$ so that it satisfies the required conditions (submodularity and optionally monotonicity). For that purpose, we add a suitable multiple of the following function:

$$J(x) = |X|^2 + 3|X|\sum_{i \in X} x_i - \left(\sum_{i \in X} x_i\right)^2.$$

We have $0 \leq J(x) \leq 3|X|^2$, $\frac{\partial J}{\partial x_i} = 3|X| - 2\sum_{i \in X} x_i \geq |X|$. Further, $\frac{\partial^2 J}{\partial x_i \partial x_j} = -2$. To make $\tilde{F}(x)$ submodular and optionally monotone, we define:

$$\hat{F}(x) = \tilde{F}(x) + 256M|X|\alpha J(x),$$
$$\hat{G}(x) = G(x) + 256M|X|\alpha J(x).$$

We verify the properties of $\hat{F}(x)$ and $\hat{G}(x)$:

- Due to Lemma 3.6, $\phi(t) = 1$ for $t \in [0, \delta]$. Hence, whenever $D(x) = ||x - \bar{x}||^2 \leq \delta$, we have $\tilde{F}(x) = G(x) = F(\bar{x})$, which depends only on $\bar{x}$. Also, we have $\hat{F}(x) = \hat{G}(x) = F(\bar{x}) + 256M|X|\alpha J(x)$ and again, $J(x)$ depends only on $\bar{x}$ (in fact, only on the average of all coordinates of x). Therefore, $\hat{F}(x)$ and $\hat{G}(x)$ in this case depend only on $\bar{x}$.
- Since $D(x^*) = \beta$, Lemma 3.6 guarantees that $0 \leq \phi(D(x^*)) < e^{-1/\alpha}$ and

$$\begin{aligned}\hat{F}(x^*) &= (1 - \phi(D(x^*))F(x^*) + \phi(D(x^*))G(x) \\ &\quad +256M|X|\alpha J(x) \\ &\geq (1 - e^{-1/\alpha})F(x^*) + 256M|X|^3\alpha \\ &\geq F(x^*) = OPT\end{aligned}$$

  using $G(x) \geq 0$ and $0 \leq F(x^*) \leq M$.
- For any $x \in P(\mathcal{F})$, we have

$$\begin{aligned}\hat{G}(x) &= G(x) + 256M|X|\alpha J(x) \\ &\leq G(x) + 768M|X|^3\alpha \leq (1 + \epsilon)\overline{OPT}.\end{aligned}$$

- Assuming $\frac{\partial F}{\partial x_i} \geq 0$, we get $\frac{\partial \tilde{F}}{\partial x_i} \geq -64M|X|\alpha$ by Lemma 3.7. Using $\frac{\partial J}{\partial x_i} \geq |X|$, we get $\frac{\partial \hat{F}}{\partial x_i} = \frac{\partial \tilde{F}}{\partial x_i} + 256M|X|\alpha J(x) \geq 0$. The same holds for $\frac{\partial \hat{G}}{\partial x_i}$ since $\frac{\partial G}{\partial x_i} \geq 0$.
- Assuming $\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$, we get $\frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} \leq 512M|X|\alpha$ by Lemma 3.7. Using $\frac{\partial^2 J}{\partial x_i \partial x_j} = -2$, we get $\frac{\partial^2 \hat{F}}{\partial x_i \partial x_j} = \frac{\partial^2 \tilde{F}}{\partial x_i \partial x_j} + 256M|X|\alpha \frac{\partial^2 J}{\partial x_i \partial x_j} \leq 0$. The same holds for $\frac{\partial^2 \hat{G}}{\partial x_i \partial x_j}$ since $\frac{\partial^2 G}{\partial x_i \partial x_j} \leq 0$.

■

This concludes the proof of our main hardness result.

4. Algorithms using the multilinear relaxation

Here we turn to our algorithmic results. First, we discuss the problem of maximizing a submodular (but not necessarily monotone) function subject to a matroid independence constraint.

4.1. Matroid independence constraint

Consider the problem $\max\{f(S) : S \in \mathcal{I}\}$, where $\mathcal{I}$ is the collection of independent sets in a matroid $\mathcal{M}$. We design an algorithm based on the multilinear relaxation of the problem, $\max\{F(x) : x \in P(\mathcal{M})\}$. Our algorithm can be seen as "continuous local search" in the matroid polytope $P(\mathcal{M})$, constrained in addition by the box $[0,t]^X$ for some fixed $t \in [0,1]$. The intuition is that this forces our local search to use fractional solutions that are more fuzzy than integral solutions and therefore less likely to get stuck in a local optimum. On the other hand, restraining the search space too much would not give us much freedom in searching for a good fractional point. This leads to a tradeoff and an optimal choice of $t \in [0,1]$ which we leave for later.

The matroid polytope is defined as $P(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_I : I \in \mathcal{I}\}$. We define

$$P_t(\mathcal{M}) = P(\mathcal{M}) \cap [0,t]^X = \{x \in P(\mathcal{M}) : \forall i; x_i \leq t\}.$$

We consider the problem $\max\{F(x) : x \in P_t(\mathcal{M})\}$. We remind the reader that $F(x) = \mathbf{E}[f(\hat{x})]$ denotes the multilinear extension. Our algorithm works as follows.

Fractional local search in $P_t(\mathcal{M})$
(given $t = \frac{r}{q}$, $r \leq q$ integer)

1) Start with $x := (0,0,\ldots,0)$. Fix $\delta = 1/q$.
2) If there is $i,j \in X$ and a direction $\mathbf{v} \in \{\mathbf{e}_j, -\mathbf{e}_i, \mathbf{e}_j - \mathbf{e}_i\}$ such that $x+\delta\mathbf{v} \in P_t(\mathcal{M})$ and $F(x+\delta\mathbf{v}) > F(x)$, set $x := x + \delta\mathbf{v}$ and repeat.
3) If there is no such direction $\mathbf{v}$, apply pipage rounding to x and return the resulting solution.

Notes. The procedure as presented here would not run in polynomial time. A modification which runs in polynomial time is that we move to a new solution only if $F(x+\delta\mathbf{v}) > F(x) + \frac{\delta}{poly(n)} OPT$ (where we first get a rough estimate of OPT using previous methods). For simplicity, we analyze the variant above and finally discuss why we can modify it without losing too much in the approximation factor. We also defer the question of how to estimate the value of $F(x)$ to the end of this section.

For $t = 1$, we have $\delta = 1$ and the procedure reduces to discrete local search. However, it is known that discrete local search alone does not give any approximation guarantee. With additional modifications, an algorithm based on discrete local search achieves a $(\frac{1}{4} - o(1))$-approximation [14].

Our version of fractional local search avoids this issue and leads directly to a good fractional solution. Throughout the algorithm, we maintain x as a linear combination of q independent sets such that no element appears in more than r of them. A local step corresponds to an add/remove/switch operation preserving this condition.

Finally, we use pipage rounding to convert a fractional solution into an integral one. As we show in Lemma A.8, a modification of the technique from [3] can be used to find an integral solution without any loss in the objective function.

Theorem 4.1. *The fractional local search algorithm for any fixed $t \in [0, \frac{1}{2}(3-\sqrt{5})]$ returns a solution of value at least $(t - \frac{1}{2}t^2)OPT$, where $OPT = \max\{f(S) : S \in \mathcal{I}\}$.*

We remark that for $t = \frac{1}{2}(3-\sqrt{5})$, we would obtain a $\frac{1}{4}(-1+\sqrt{5}) \doteq 0.309$-approximation, improving the factor of $\frac{1}{4}$ [14]. This is not a rational value, but we can pick a rational t arbitrarily close to $\frac{1}{2}(3-\sqrt{5})$. For values $t > \frac{1}{2}(3-\sqrt{5})$, our analysis does not yield a better approximation factor.

First, we discuss properties of the point found by the fractional local search algorithm.

Lemma 4.2. *The outcome of the fractional local search algorithm x is a "fractional local optimum" in the following sense. (All the partial derivatives are evaluated at x.)*

- *For any i such that $x - \delta\mathbf{e}_i \in P_t(\mathcal{M})$, $\frac{\partial F}{\partial x_i} \geq 0$.*
- *For any j such that $x + \delta\mathbf{e}_j \in P_t(\mathcal{M})$, $\frac{\partial F}{\partial x_j} \leq 0$.*
- *For any i,j such that $x + \delta(\mathbf{e}_j - \mathbf{e}_i) \in P_t(\mathcal{M})$, $\frac{\partial F}{\partial x_j} - \frac{\partial F}{\partial x_i} \leq 0$.*

Proof: We use the property (see [3]) that along any direction $\mathbf{v} = \pm\mathbf{e}_i$ or $\mathbf{v} = \mathbf{e}_i - \mathbf{e}_j$, the function $F(x + \lambda\mathbf{v})$ is a convex function of λ. Also, observe that if it is possible to move from x in the direction of $\mathbf{v}$ by any nonzero amount, then it is possible to move by $\delta\mathbf{v}$, because all coordinates of x are integer multiples of δ and all the constraints also have coefficients which are integer multiples of δ. Therefore, if $\frac{dF}{d\lambda} > 0$ and it is possible to move in the direction of $\mathbf{v}$, we would get $F(x+\delta\mathbf{v}) > F(x)$ and the fractional local search would continue.

If $\mathbf{v} = -\mathbf{e}_i$ and it is possible to move along $-\mathbf{e}_i$, we get $\frac{dF}{d\lambda} = -\frac{\partial F}{\partial x_i} \leq 0$. Similarly, if $\mathbf{v} = \mathbf{e}_j$ and it is possible to move along $\mathbf{e}_j$, we get $\frac{dF}{d\lambda} = \frac{\partial F}{\partial x_j} \leq 0$. Finally, if $\mathbf{v} = \mathbf{e}_j - \mathbf{e}_i$ and it is possible to move along $\mathbf{e}_j - \mathbf{e}_i$, we get

$\frac{dF}{d\lambda} = \frac{\partial F}{\partial x_j} - \frac{\partial F}{\partial x_i} \leq 0.$ ∎

In the following, we refer to the following exchange property for matroids (which follows easily from [22], Corollary 39.12a; see also [14]).

Lemma 4.3. *If $I, C \in \mathcal{I}$, then for any $j \in C \setminus I$, there is $\pi(j) \subseteq I \setminus C$, $|\pi(j)| \leq 1$, such that $I \setminus \pi(j) + j \in \mathcal{I}$. Moreover, the sets $\pi(j)$ are disjoint (each $i \in I \setminus C$ appears at most once as $\pi(j) = \{i\}$).*

Using this, we prove a lemma about fractional local optima which generalizes Lemma 2.2 in [14].

Lemma 4.4. *Let x be the outcome of fractional local search over $P_t(\mathcal{M})$. Let $C \in \mathcal{I}$ be any independent set. Let $C' = \{i \in C : x_i < t\}$. Then*

$$2F(x) \geq F(x \vee \mathbf{1}_{C'}) + F(x \wedge \mathbf{1}_C).^5$$

Note that for $t = 1$, the lemma reduces to $2F(x) \geq F(x \vee \mathbf{1}_C) + F(x \wedge \mathbf{1}_C)$ (similar to Lemma 2.2 in [14]). For $t < 1$, however, it is necessary to replace C by C' in the first expression, which becomes apparent in the proof. The reason is that we do not have any information on $\frac{\partial F}{\partial x_i}$ for coordinates where $x_i = t$.

Proof: Let $C \in \mathcal{I}$ and assume $x \in P_t(\mathcal{M})$ is a local optimum. We can decompose it into a convex linear combination of vertices of $P(\mathcal{M})$, $x = \sum_{I \in \mathcal{I}} x_I \mathbf{1}_I$ where $\sum x_I = 1$. By the smooth submodularity of $F(x)$ (see [24]),

$$F(x \vee \mathbf{1}_{C'}) - F(x) \leq \sum_{j \in C'} (1 - x_j) \frac{\partial F}{\partial x_j}$$
$$= \sum_{j \in C'} \sum_{I : j \notin I} x_I \frac{\partial F}{\partial x_j} = \sum_I x_I \sum_{j \in C' \setminus I} \frac{\partial F}{\partial x_j}.$$

All partial derivatives here are evaluated at x. On the other hand, also by submodularity,

$$F(x) - F(x \wedge \mathbf{1}_C) \geq \sum_{i \notin C} x_i \frac{\partial F}{\partial x_i}$$
$$= \sum_{i \notin C} \sum_{I : i \in I} x_I \frac{\partial F}{\partial x_i} = \sum_I x_I \sum_{i \in I \setminus C} \frac{\partial F}{\partial x_i}.$$

To prove the lemma, it remains to prove the following.

Claim. Whenever $x_I > 0$, $\sum_{j \in C' \setminus I} \frac{\partial F}{\partial x_j} \leq \sum_{i \in I \setminus C} \frac{\partial F}{\partial x_i}$.

Proof: For any $I \in \mathcal{I}$, we can apply Lemma 4.3 to get a mapping π such that $I \setminus \pi(j) + j \in \mathcal{I}$ for any $j \in C \setminus I$. Now, consider $j \in C' \setminus I$, i.e. $j \in C \setminus I$ and $x_j < t$.

If $\pi(j) = \emptyset$, is possible to move from x in the direction of $\mathbf{e}_j$, because $I + j \in \mathcal{I}$ and hence we can replace I by $I + j$ (or at least we can do this for some nonzero fraction of its coefficient) in the linear combination. Because $x_j < t$, we can move by a nonzero amount inside $P_t(\mathcal{M})$. By Lemma 4.2, $\frac{\partial F}{\partial x_j} \leq 0$.

Similarly, if $\pi(j) = \{i\}$, it is possible to move in the direction of $\mathbf{e}_j - \mathbf{e}_i$, because I can be replaced by $I \setminus \pi(j) + i$ for some nonzero fraction of its coefficient. By Lemma 4.2, in this case $\frac{\partial F}{\partial x_j} - \frac{\partial F}{\partial x_i} \leq 0$.

Finally, for any $i \in I$ we have $x_i > 0$ and therefore we can decrease x_i while staying inside $P_t(\mathcal{M})$. By Lemma 4.2, we have $\frac{\partial F}{\partial x_i} \geq 0$ for all $i \in I$. This means

$$\sum_{j \in C' \setminus I} \frac{\partial F}{\partial x_j} \leq \sum_{j \in C' \setminus I : \pi(j) = \{i\}} \frac{\partial F}{\partial x_i} \leq \sum_{i \in I \setminus C} \frac{\partial F}{\partial x_i}$$

using the inequalities we derived above, and the fact that each $i \in I \setminus C$ appears at most once in $\pi(j)$. This proves the Claim, and hence the Lemma. ∎

Now we are ready to prove Theorem 4.1.

Proof: Let x be the outcome of the fractional local search over $P_t(\mathcal{M})$. Define $A = \{i : x_i = t\}$. Let C be the optimum solution and $C' = C \setminus A = \{i \in C : x_i < t\}$. By Lemma 4.4,

$$2F(x) \geq F(x \vee \mathbf{1}_{C'}) + F(x \wedge \mathbf{1}_C).$$

First, let's analyze $F(x \wedge \mathbf{1}_C)$. We apply Lemma A.4, which states that $F(x \wedge \mathbf{1}_C) \geq \mathbf{E}[f(T(x \wedge \mathbf{1}_C))]$. Here, $T(x \wedge \mathbf{1}_C)$ is a random threshold set corresponding to the vector $x \wedge \mathbf{1}_C$, i.e.

$$T(x \wedge \mathbf{1}_C) = \{i \in C : x_i > \lambda\} = T(x) \cap C.$$

Therefore,

$$F(x \wedge \mathbf{1}_C) \geq \mathbf{E}[f(T(x) \cap C)].$$

Due to the definition of a threshold set, with probability t we have $\lambda < t$ and $T(x)$ contains $A = \{i : x_i = t\} = C \setminus C'$. Then, $f(T(x) \cap C) + f(C') \geq f(C)$ by submodularity. We conclude that

$$F(x \wedge \mathbf{1}_C) \geq t(f(C) - f(C')). \quad (4)$$

Next, let's analyze $F(x \vee \mathbf{1}_{C'})$. We consider the ground set partitioned into $X = C \cup \bar{C}$, and we apply Lemma A.5. We get

$$F(x \vee \mathbf{1}_{C'}) \geq \mathbf{E}[f((T_1(x \vee \mathbf{1}_{C'}) \cap C) \cup (T_2(x \vee \mathbf{1}_{C'}) \cap \bar{C}))].$$

The random threshold sets look as follows: $T_1(x \vee \mathbf{1}_{C'}) \cap C = (T_1(x) \cup C') \cap C$ is equal to C with probability t, and equal to C' otherwise. $T_2(x \vee \mathbf{1}_{C'}) \cap \bar{C} = T_2(x) \cap \bar{C}$ is empty with probability $1 - t$. (We ignore the contribution when $T_2(x) \cap \bar{C} \neq \emptyset$.) Because T_1 and T_2 are independently sampled, we get

$$F(x \vee \mathbf{1}_{C'}) \geq t(1-t) f(C) + (1-t)^2 f(C').$$

Provided that $t \in [0, \frac{1}{2}(3 - \sqrt{5})]$, we have $t \leq (1-t)^2$. Then, we can write

$$F(x \vee \mathbf{1}_{C'}) \geq t(1-t) f(C) + t f(C'). \quad (5)$$

Combining equations (4) and (5), we get

$$F(x \vee \mathbf{1}_{C'}) + F(x \wedge \mathbf{1}_C) \geq t(f(C) - f(C'))$$
$$+ t(1-t) f(C) + t\, f(C') = (2t - t^2) f(C).$$

[5] $x \vee y$ denotes the coordinate-wise maximum, $(x \vee y)_i = \max\{x_i, y_i\}$. $x \wedge y$ denotes the coordinate-minimum, $(x \wedge y)_i = \min\{x_i, y_i\}$.

Therefore,

$$F(x) \geq \frac{1}{2}(F(x \vee \mathbf{1}_{C'}) + F(x \wedge \mathbf{1}_C)) \geq (t - \frac{1}{2}t^2)f(C).$$

Finally, we apply the pipage rounding technique which does not lose anything in terms of objective value (see Lemma A.8). ■

Technical remarks: In each step of the algorithm, we need to estimate values of $F(x)$ for given $x \in P_t(\mathcal{M})$. We accomplish this by using the expression $F(x) = \mathbf{E}[f(R(x))]$ where $R(x)$ is a random set associated with x. By standard bounds, if the values of $f(S)$ are in a range $[0, M]$, we can achieve accuracy $M/poly(n)$ using a polynomial number of samples. We use the fact that $OPT \geq \frac{1}{n}M$ (see Lemma A.9 in the appendix) and therefore we can achieve $OPT/poly(n)$ additive error in polynomial time.

We also relax the local step condition: we move to the next solution only if $F(x + \delta\mathbf{v}) > F(x) + \frac{\delta}{poly(n)}OPT$ for a suitable polynomial in n. This way, we can only make a polynomial number of steps. When we terminate, the local optimality conditions (Lemma 4.2) are satisfied within an additive error of $OPT/poly(n)$, which yields a polynomially small error in the approximation bound.

4.2. Matroid base constraint

Let us move on to the problem $\max\{f(S) : S \in \mathcal{B}\}$ where $\mathcal{B}$ are the bases of a matroid. For a fixed $t \in [0, 1]$, let us consider an algorithm which can be seen as local search inside the base polytope $B(\mathcal{M})$, further constrained by the box $[0, t]^X$. The matroid base polytope is defined as $B(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_B : B \in \mathcal{B}\}$ or equivalently [5] as $B(\mathcal{M}) = \{x \geq 0 : \forall S; \sum_{i \in S} x_i \leq r_{\mathcal{M}}(S), \sum_{i \in X} x_i = r_{\mathcal{M}}(X)\}$, where $r_{\mathcal{M}}$ is the matroid rank function of $\mathcal{M}$. Finally, we define

$$B_t(\mathcal{M}) = B(\mathcal{M}) \cap [0, t]^X = \{x \in B(\mathcal{M}) : \forall i; x_i \leq t\}.$$

Observe that $B_t(\mathcal{M})$ is nonempty if and only if there is a convex linear combination $x = \sum_{B \in \mathcal{B}} \xi_B \mathbf{1}_B$ such that $x_i \in [0, t]$ for all i. This is equivalent to saying that there is a linear combination $x' = \sum_{B \in \mathcal{B}} \xi'_B \mathbf{1}_B$ such that $x_i \in [0, 1]$ and $\sum \xi'_B = 1/t$, in other words the fractional base packing number is $\nu \geq 1/t$. Since the optimal fractional packing of bases can be found efficiently [22], we can find efficiently the minimum $t \in [\frac{1}{2}, 1]$ such that $B_t(\mathcal{M}) \neq \emptyset$. Then, our algorithm is the following.

Fractional local search in $B_t(\mathcal{M})$
(given $t = \frac{r}{q}$, $r \leq q$ integer)

1) Let $\delta = 1/q$. Assume that $x \in B_t(\mathcal{M})$ and the coordinates of x are integer multiples of δ.
2) If there is a direction $\mathbf{v} = \mathbf{e}_j - \mathbf{e}_i$ such that $x + \delta\mathbf{v} \in B_t(\mathcal{M})$ and $F(x + \delta\mathbf{v}) > F(x)$, then set $x := x + \delta\mathbf{v}$ and repeat.
3) If there is no such direction $\mathbf{v}$, apply pipage rounding to x and return the resulting solution.

Notes. We remark that the starting point can be found as a convex linear combination of q bases, $x = \frac{1}{q}\sum_{i=1}^{q} \mathbf{1}_{B_i}$, such that no element appears in more than r of them, using matroid union techniques [22]. In the algorithm, we maintain this representation. The local search step corresponds to switching a pair of elements in one base, under the condition that no element is used in more than r bases at the same time. For now, we ignore the issues of estimating $F(x)$ and stopping the local search within polynomial time. We discuss this at the end of this section.

Finally, we use pipage rounding to convert the fractional solution x into an integral one of value at least $F(x)$ (Lemma A.6). Note that it is not necessarily true that any of the bases in a convex linear combination $x = \sum \xi_B \mathbf{1}_B$ achieves the value $F(x)$.

Theorem 4.5. *If there is a fractional packing of $\nu \in [1, 2]$ bases in $\mathcal{M}$, then the fractional local search algorithm with $t = 1/\nu$ returns a solution of value at least $\frac{1}{2}(1 - t)$ OPT.*

For example, assume that $\mathcal{M}$ contains two disjoint bases B_1, B_2 (which is the case considered in [14]). Then, the algorithm can be used with $t = 1/2$ and and we obtain a $(1/4 - o(1))$-approximation, improving the $(1/6 - o(1))$-approximation from [14]. If there is a fractional packing of more than 2 bases, our analysis still gives only a $(1/4 - o(1))$-approximation. If the dual matroid $\mathcal{M}^*$ admits a better fractional packing of bases, we can consider the problem $\max\{f(\bar{S}) : S \in \mathcal{B}^*\}$ which is equivalent. For a uniform matroid, $\mathcal{B} = \{B : |B| = k\}$, the fractional base packing number is either at least 2 or the same holds for the dual matroid, $\mathcal{B}^* = \{B : |B| = n - k\}$ (as noted in [14]). Therefore, we get a $(1/4 - o(1))$-approximation for any uniform matroid. The value $t = 1$ can be used for any matroid, but it does not yield any approximation guarantee.

Analysis of the algorithm: We turn to the properties of fractional local optima. We will prove that the point x found by the fractional local search algorithm satisfies the following conditions that allow us to compare $F(x)$ to the actual optimum.

Lemma 4.6. *The outcome of the fractional local search algorithm x is a "fractional local optimum" in the following sense.*

- *For any i, j such that $x + \delta(\mathbf{e}_j - \mathbf{e}_i) \in B_t(\mathcal{M})$,*

$$\frac{\partial F}{\partial x_j} - \frac{\partial F}{\partial x_i} \leq 0.$$

(The partial derivatives are evaluated at x.)

Proof: Observe that the coordinates of x are always integer multiples of δ, therefore if it is possible to move from x in the direction of $\mathbf{v} = \mathbf{e}_j - \mathbf{e}_i$ by any nonzero amount, then it is possible to move by $\delta\mathbf{v}$. We use the property that for any direction $\mathbf{v} = \mathbf{e}_j - \mathbf{e}_i$, the function $F(x + \lambda\mathbf{v})$ is a convex function of λ [3]. Therefore, if $\frac{dF}{d\lambda} > 0$ and it is possible to move in the direction of $\mathbf{v}$, we would get $F(x+\delta\mathbf{v}) > F(x)$ and the fractional local search would continue. For $\mathbf{v} = \mathbf{e}_j - \mathbf{e}_i$, we get

$$\frac{dF}{d\lambda} = \frac{\partial F}{\partial x_j} - \frac{\partial F}{\partial x_i} \leq 0.$$

■

In the following, we refer to the following exchange property for matroid bases (see [22], Corollary 39.21a).

Lemma 4.7. *For any $B_1, B_2 \in \mathcal{B}$, there is a bijection $\pi : B_1 \setminus B_2 \to B_2 \setminus B_1$ such that $\forall i \in B_1 \setminus B_2$; $B_1 - i + \pi(i) \in \mathcal{M}$.*

Using this, we prove a lemma about fractional local optima analogous to Lemma 2.2 in [14].

Lemma 4.8. *Let x be the outcome of fractional local search over $B_t(\mathcal{M})$. Let $C \in \mathcal{B}$ be any base. Then there is $c \in [0,1]^X$ satisfying*

- *$c_i = t$, if $i \in C$ and $x_i = t$*
- *$c_i = 1$, if $i \in C$ and $x_i < t$*
- *$0 \leq c_i \leq x_i$, if $i \notin C$*

such that

$$2F(x) \geq F(x \vee c) + F(x \wedge c).$$

Note that for $t = 1$, we can set $c = \mathbf{1}_C$. However, in general we need this more complicated formulation. Intuitively, c is obtained from x by raising the variables $x_i, i \in C$ and decreasing them for $i \notin C$. However, we can only raise the variables $x_i, i \in C$, where x_i is below the threshold t, otherwise we do not have any information about $\frac{\partial F}{\partial x_i}$. Also, we do not necessarily decrease all the variables outside of C to zero.

Proof: Let $C \in \mathcal{B}$ and assume $x \in B_t(\mathcal{M})$ is a fractional local optimum. We can decompose x into a convex linear combination of vertices of $B(\mathcal{M})$, $x = \sum \xi_B \mathbf{1}_B$. By Lemma 4.7, for each base B there is a bijection $\pi_B : B \setminus C \to C \setminus B$ such that $\forall i \in B \setminus C$; $B - i + \pi_B(i) \in \mathcal{B}$.

We define $C' = \{i \in C : x_i < t\}$. The reason we consider C' is that if $x_i = t$, there is no room for an exchange step increasing x_i, and therefore Lemma 4.6 does not give any information about $\frac{\partial F}{\partial x_i}$. We construct the vector c by starting from x, and for each B swapping the elements in $B \setminus C$ for their image under π_B, provided it is in C', until we raise the coordinates on C' to $c_i = 1$. Formally, we set $c_i = 1$ for $i \in C'$, $c_i = t$ for $i \in C \setminus C'$, and for each $i \notin C$, we define

$$c_i = x_i - \sum_{B: i \in B, \pi_B(i) \in C'} \xi_B.$$

In the following, all partial derivatives are evaluated at x. By the smooth submodularity of $F(x)$ (see [24]),

$$F(x \vee c) - F(x) \quad \leq \quad \sum_{j: c_j > x_j} (c_j - x_j) \frac{\partial F}{\partial x_j}$$

$$= \sum_{j \in C'} (1 - x_j) \frac{\partial F}{\partial x_j} = \sum_B \sum_{j \in C' \setminus B} \xi_B \frac{\partial F}{\partial x_j} \tag{6}$$

because $\sum_{B: j \notin B} \xi_B = 1 - x_j$ for any j. On the other hand, also by smooth submodularity,

$$F(x) - F(x \wedge c) \quad \geq \quad \sum_{i: c_i < x_i} (x_i - c_i) \frac{\partial F}{\partial x_i}$$

$$= \sum_{i \notin C} (x_i - c_i) \frac{\partial F}{\partial x_i} = \sum_{i \notin C} \sum_{B: i \in B, \pi_B(i) \in C'} \xi_B \frac{\partial F}{\partial x_i}$$

using our definition of c_i. In the last sum, for any nonzero contribution, we have $\xi_B > 0$, $i \in B$ and $j = \pi_B(i) \in C'$, i.e. $x_j < t$. Therefore it is possible to move in the direction $\mathbf{e}_j - \mathbf{e}_i$ (we can switch from B to $B - i + j$). By Lemma 4.6,

$$\frac{\partial F}{\partial x_j} - \frac{\partial F}{\partial x_i} \leq 0.$$

Therefore, we get

$$F(x) - F(x \wedge c) \quad \geq \quad \sum_{i \notin C} \sum_{B: i \in B, j = \pi_B(i) \in C'} \xi_B \frac{\partial F}{\partial x_j}$$

$$= \sum_B \sum_{i \in B \setminus C: j = \pi_B(i) \in C'} \xi_B \frac{\partial F}{\partial x_j}. \tag{7}$$

By the bijective property of π_B, this is equal to $\sum_B \sum_{j \in C' \setminus B} \xi_B \frac{\partial F}{\partial x_j}$. Putting (6) and (7) together, we get $F(x \vee c) - F(x) \leq F(x) - F(x \wedge c)$. ■

Now we are ready to prove Theorem 4.5.

Proof: Assuming that $B_t(\mathcal{M}) \neq \emptyset$, we can find a starting point $x^0 \in B_t(\mathcal{M})$. From this point, we reach a fractional local optimum $x \in B_t(\mathcal{M})$ (see Lemma 4.6). We want to compare $F(x)$ to the actual optimum; assume that $OPT = f(C)$.

As before, we define $C' = \{i \in C : x_i < t\}$. By Lemma 4.8, we know that the fractional local optimum satisfies:

$$2F(x) \geq F(x \vee c) + F(x \wedge c) \tag{8}$$

for some vector c such that $c_i = t$ for all $i \in C \setminus C'$, $c_i = 1$ for $i \in C'$ and $0 \leq c_i \leq x_i$ for $i \notin C$.

First, let's analyze $F(x \vee c)$. We have

- $(x \vee c)_i = 1$ for all $i \in C'$.
- $(x \vee c)_i = t$ for all $i \in C \setminus C'$.
- $(x \vee c)_i \leq t$ for all $i \notin C$.

We apply Lemma A.5 to the partition $X = C \cup \bar{C}$. We get

$$F(x \vee c) \geq \mathbf{E}[f((T_1(x \vee c) \cap C) \cup (T_2(x \vee c) \cap \bar{C}))]$$

where $T_1(x)$ and $T_2(x)$ are independent threshold sets. Based on the information above, $T_1(x \vee c) \cap C = C$ with probability t and $T_1(x \vee c) \cap C = C'$ otherwise. On the other hand, $T_2(x \vee c) \cap \bar{C} = \emptyset$ with probability at least $1 - t$. These two events are independent. We conclude that on the right-hand side, we get $f(C)$ with probability at least $t(1-t)$, or $f(C')$ with probability at least $(1-t)^2$:

$$F(x \vee c) \geq t(1-t)f(C) + (1-t)^2 f(C'). \tag{9}$$

Turning to $F(x \wedge c)$, we see that

- $(x \wedge c)_i = x_i$ for all $i \in C'$.
- $(x \wedge c)_i = t$ for all $i \in C \setminus C'$.
- $(x \wedge c)_i \leq t$ for all $i \notin C$.

We apply Lemma A.5 to $X = C \cup \bar{C}$.

$$F(x \wedge c) \geq \mathbf{E}[f((T_1(x \wedge c) \cap C) \cup (T_2(x \wedge c) \cap \bar{C}))].$$

With probability t, $T_1(x \wedge c) \cap C$ contains at least $C \setminus C'$ (and maybe some elements of C'). In this case, $f(T_1(x \wedge c) \cap C) \geq f(C) - f(C')$ by submodularity. Also, $T_2(x \wedge c) \cap \bar{C}$ is empty with probability at least $1-t$. Again, these two events are independent. Therefore, $F(x \wedge c) \geq t(1-t)(f(C) - f(C'))$. If $f(C') > f(C)$, this bound is vacuous; otherwise, we can replace $t(1-t)$ by $(1-t)^2$, because $t \geq 1/2$. In any case,

$$F(x \wedge c) \geq (1-t)^2 (f(C) - f(C')). \tag{10}$$

Combining (8), (9) and (10),

$$F(x) \geq \frac{1}{2}(F(x \vee c) + F(x \wedge c))$$

$$\geq \frac{1}{2}(t(1-t)f(C) + (1-t)^2 f(C)) = \frac{1}{2}(1-t)f(C).$$

■

Technical remarks: Again, we have to deal with the issues of estimating $F(x)$ and stopping the local search in polynomial time. We do this exactly as we did at the end of Section 4.1. One issue to be careful about here is that if $f : 2^X \to [0, M]$, our estimates of $F(x)$ have an additive error of $M/poly(n)$. If the optimum value $OPT = \max\{f(S) : S \in \mathcal{B}\}$ is very small compared to M, the error might be large compared to OPT which would be a problem. The optimum could in fact be very small in general. But it holds that if $\mathcal{M}$ contains no loops and co-loops (which can be eliminated easily), then $OPT \geq \frac{1}{n^2} M$ (see Appendix C). Then, our sampling errors are on the order of $OPT/poly(n)$ which yields a $1/poly(n)$ error in the approximation bound.

5. Approximation for symmetric instances

We can achieve a better approximation assuming that the instance exhibits a certain symmetry. This is the same kind of symmetry that we use in our hardness construction (Section 3) and the hard instances exhibit the same symmetry as well. It turns out that our approximation in this case matches the hardness threshold up to lower order terms. Hence, we can say that the case of symmetric instances is now completely understood.

Similar to our hardness result, the symmetries that we consider here are permutations of the ground set X, corresponding to permutations of coordinates in $\mathbb{R}^X$. We start with some basic properties which are helpful in analyzing symmetric instances.

Lemma 5.1. *Assume that $f : 2^X \to \mathbb{R}$ is invariant with respect to a group of permutations $\mathcal{G}$ and $F(x) = \mathbf{E}[f(\hat{x})]$. Then for any symmetrized vector $\bar{c} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(c)]$, $\nabla F|_{\bar{c}}$ is also symmetric w.r.t. $\mathcal{G}$. I.e., for any $\tau \in \mathcal{G}$,*

$$\tau(\nabla F|_{x=\bar{c}}) = \nabla F|_{x=\bar{c}}.$$

Proof: Since $f(S)$ is invariant under $\mathcal{G}$, so is $F(x)$, i.e. $F(x) = F(\tau(x))$ for any $\tau \in \mathcal{G}$. Differentiating both sides at $x = c$, we get by the chain rule:

$$\frac{\partial F}{\partial x_i}\Big|_{x=c} = \sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\tau(c)} \frac{\partial}{\partial x_i}(\tau(x))_j = \sum_j \frac{\partial F}{\partial x_j}\Big|_{x=\tau(c)} \frac{\partial x_{\tau(j)}}{\partial x_i}.$$

Here, $\frac{\partial x_{\tau(j)}}{\partial x_i} = 1$ if $\tau(j) = i$, and 0 otherwise. Therefore,

$$\frac{\partial F}{\partial x_i}\Big|_{x=c} = \frac{\partial F}{\partial x_{\tau^{-1}(i)}}\Big|_{x=\tau(c)}.$$

Note that $\tau(\bar{c}) = \mathbf{E}_{\sigma \in \mathcal{G}}[\tau(\sigma(c))] = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(c)] = \bar{c}$ since the distribution of $\tau \circ \sigma$ is equal to the distribution of σ. Therefore,

$$\frac{\partial F}{\partial x_i}\Big|_{x=\bar{c}} = \frac{\partial F}{\partial x_{\tau^{-1}(i)}}\Big|_{x=\bar{c}}$$

for any $\tau \in \mathcal{G}$. ■

Next, we prove that the "symmetric optimum" $\max\{F(\bar{x}) : x \in P(\mathcal{F})\}$ gives a solution which is a local optimum for the original instance $\max\{F(x) : x \in P(\mathcal{F})\}$. (As we proved in Section 3, in general we cannot hope to find a better solution than the symmetric optimum.)

Lemma 5.2. *Let $f : 2^X \to \mathbb{R}$ and $\mathcal{F} \subset 2^X$ be invariant with respect to a group of permutations $\mathcal{G}$. Let $\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{F})\}$ where $\bar{x} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$, and let x_0 be the symmetric point where $\overline{OPT}$ is attained ($\bar{x}_0 = x_0$). Then x_0 is a local optimum for the problem $\max\{F(x) : x \in P(\mathcal{F})\}$, in the sense that $(x - x_0) \cdot \nabla F|_{x_0} \leq 0$ for any $x \in P(\mathcal{F})$.*

Proof: Assume for the sake of contradiction that $(x - x_0) \cdot \nabla F|_{x_0} > 0$ for some $x \in P(\mathcal{F})$. We use the symmetric properties of f and $\mathcal{F}$ to show that $(\bar{x} - x_0) \cdot \nabla F|_{x_0} > 0$ as well. Recall that $x_0 = \bar{x}_0$. We have

$$(\bar{x} - x_0) \cdot \nabla F|_{x_0} = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(x - x_0) \cdot \nabla F|_{x_0}]$$

$$= \mathbf{E}_{\sigma \in \mathcal{G}}[(x - x_0) \cdot \sigma^{-1}(\nabla F|_{x_0})] = (x - x_0) \cdot \nabla F|_{x_0} > 0$$

using Lemma 5.1. Hence, there would be a direction $\bar{x} - x_0$ along which an improvement can be obtained. But then, consider a small $\delta > 0$ such that $x_1 = x_0 + \delta(\bar{x} - x_0) \in P(\mathcal{F})$ and also $F(x_1) > F(x_0)$. The point x_1 is symmetric ($\bar{x}_1 = x_1$) and hence it would contradict the assumption that $F(x_0) = \overline{OPT}$. ■

5.1. Submodular maximization over independent sets

Let us derive an optimal approximation result for the problem $\max\{f(S) : S \in \mathcal{I}\}$ under the assumption that the instance is "element-transitive". This means that there is a group of permutations $\mathcal{G}$ such that the orbit of any element is the entire ground set X, and our instance is invariant under $\mathcal{G}$. Then, we show that it is easy to achieve an optimal $(\frac{1}{2} - o(1))$-approximation.

Theorem 5.3. *Let $\max\{f(S) : S \in \mathcal{I}\}$ be an instance symmetric with respect to an element-transitive group of permutations $\mathcal{G}$. Let $\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{M})\}$ where $\bar{x} = \mathbf{E}_{\sigma\in\mathcal{G}}[\sigma(x)]$. Then $\overline{OPT} \geq \frac{1}{2}OPT$.*

Proof: Let $OPT = f(C)$. By Lemma 5.2, $\overline{OPT} = F(x_0)$ where x_0 is a local optimum for the problem $\max\{F(x) : x \in P(\mathcal{M})$. This means it is also a local optimum in the sense of Lemma 4.2, with $t = 1$. By Lemma 4.4,

$$F(x_0) \geq F(x_0 \vee \mathbf{1}_C) + F(x_0 \wedge \mathbf{1}_C).$$

Also, $x_0 = \bar{x}_0$. As we are dealing with an element-transitive group of symmetries, this means all the coordinates of x_0 are equal, $x_0 = (\xi, \xi, \ldots, \xi)$. Therefore, $x_0 \vee \mathbf{1}_C$ is equal to 1 on C and ξ outside of C. By Lemma A.4,

$$F(x_0 \vee \mathbf{1}_C) \geq (1 - \xi)f(C).$$

Similarly, $x_0 \wedge \mathbf{1}_C$ is equal to ξ on C and 0 outside of C. By Lemma A.4,

$$F(x_0 \wedge \mathbf{1}_C) \geq \xi f(C).$$

Combining the two bounds,

$$2F(x_0) \geq F(x_0 \vee \mathbf{1}_C) + F(x_0 \wedge \mathbf{1}_C)$$
$$\geq (1 - \xi)f(C) + \xi f(C) = f(C) = OPT.$$

■

Since all symmetric solutions $x = (\xi, \xi, \ldots, \xi)$ form a 1-parameter family, and $F(\xi, \xi, \ldots, \xi)$ is a concave function, we can search for the best symmetric solution (within any desired accuracy) by binary search. Without going into details, we get the following.

Corollary 5.4. *There is a $(\frac{1}{2} - o(1))$-approximation ("brute force" search over symmetric solutions) for the problem $\max\{f(S) : S \in \mathcal{I}\}$ for instances symmetric under an element-transitive group of permutations.*

The hard instances for submodular maximization subject to a matroid independence constraint correspond to refinements of the Max Cut instance for the graph K_2 (Section 2). It is easy to see that such instances are element-transitive, and it follows from Section 3 that a $(\frac{1}{2} + \epsilon)$-approximation for such instances would require exponentially many value queries. Therefore, our approximation for element-transitive instances is optimal.

5.2. Submodular maximization over bases

Let us come back to the problem of submodular maximization over the bases of matroid. The property that $\overline{OPT}$ is a local optimum with respect to the original problem $\max\{F(x) : x \in P(\mathcal{F})\}$ is very useful in arguing about the value of $\overline{OPT}$. We already have tools to deal with local optima from Section 4.2. Here we prove the following.

Lemma 5.5. *Let $B(\mathcal{M})$ be the matroid base polytope of $\mathcal{M}$ and $x_0 \in B(\mathcal{M})$ a local maximum for the submodular maximization problem $\max\{F(x) : x \in B(\mathcal{M})\}$, in the sense that $(x - x_0) \cdot \nabla F|_{x_0} \leq 0$ for any $x \in B(\mathcal{M})$. Assume in addition that $x_0 \in [s, t]^X$. Then*

$$F(x_0) \geq \frac{1}{2}(1 - t + s) \cdot OPT.$$

Proof: Let $OPT = \max\{f(B) : B \in \mathcal{B}\} = f(C)$. We assume that $x_0 \in B(\mathcal{M})$ is a local optimum with respect to any direction $x - x_0$, $x \in B(\mathcal{M})$, so it is also a local optimum with respect to the fractional local search in the sense of Lemma 4.8, with $t = 1$. The lemma implies that

$$2F(x_0) \geq F(x \vee \mathbf{1}_C) + F(x \wedge \mathbf{1}_C).$$

By assumption, the coordinates of $x \vee \mathbf{1}_C$ are equal to 1 on C and at most t outside of C. With probability $1 - t$, a random threshold in $[0, 1]$ falls between t and 1, and Lemma A.4 implies that

$$F(x \vee \mathbf{1}_C) \geq (1 - t) \cdot f(C).$$

Similarly, the coordinates of $x \wedge \mathbf{1}_C$ are 0 outside of C, and at least s on C. A random threshold falls between 0 and s with probability s, and Lemma A.4 implies that

$$F(x \wedge \mathbf{1}_C) \geq s \cdot f(C).$$

Putting these inequalities together, we get $2F(x_0) \geq F(x \vee \mathbf{1}_C) + F(x \wedge \mathbf{1}_C) \geq (1 - t + s) \cdot f(C)$. ■

Totally symmetric instances. The application we have in mind here is a special case of submodular maximization over the bases of a matroid, which we call *totally symmetric*.

Definition 5.6. *We call an instance $\max\{f(S) : S \in \mathcal{F}\}$ totally symmetric with respect to a group of permutations $\mathcal{G}$, if both $f(S)$ and $\mathcal{F}$ are invariant under $\mathcal{G}$ and moreover, there is a point $c \in P(\mathcal{F})$ such that $c = \bar{x} = \mathbf{E}_{\sigma\in\mathcal{G}}[\sigma(x)]$ for every $x \in P(\mathcal{F})$. We call c the center of the instance.*

Note that this is indeed stronger than just being invariant under $\mathcal{G}$. For example, an instance on a ground set $X = X_1 \cup X_2$ could be symmetric with respect to any permutation of X_1 and any permutation of X_2. For any $x \in P(\mathcal{F})$, the symmetric vector $\bar{x}$ is constant on X_1 and constant on X_2. However, in a totally symmetric instance, there should be a unique symmetric point.

Bases of partition matroids: A canonical example of a totally symmetric instance is as follows. Let $X = X_1 \cup X_2 \cup \ldots \cup X_m$ and let integers $k_1, \ldots, k_m$ be given. This defines a partition matroid $\mathcal{M} = (X, \mathcal{B})$, whose bases are

$$\mathcal{B} = \{B : \forall j; |B \cap X_j| = k_j\}.$$

The associated matroid base polytope is

$$B(\mathcal{M}) = \{x \geq 0 : \forall j; \sum_{i \in X_j} x_i = k_j\}.$$

This matroid is invariant under any group of permutations $\mathcal{G}$ which maps each X_j to itself. In particular, assume that the orbit of each element $i \in X_j$ is the entire part X_j. This implies that for any $x \in B(\mathcal{M})$, $\bar{x}$ is the same vector, with coordinates $k_j/|X_j|$ on X_j. If $f(S)$ is also invariant under $\mathcal{G}$, we have a totally symmetric instance $\max\{f(S) : S \in \mathcal{B}\}$. The center point can be found by taking any feasible solution and symmetrizing it w.r.t. $\mathcal{G}$. We show that for such instances, the center point achieves an improved approximation.

Theorem 5.7. *Let $\max\{f(S) : S \in \mathcal{B}\}$ be a totally symmetric instance. Let the fractional packing number of bases be ν and the fractional packing number of dual bases ν^*. Then the center point c satisfies*

$$F(c) \geq \left(1 - \frac{1}{2\nu} - \frac{1}{2\nu^*}\right) OPT.$$

Recall that in the general case, we get a $\frac{1}{2}(1 - 1/\nu - o(1))$-approximation (Theorem 4.5). By passing to the dual matroid, we can also obtain a $\frac{1}{2}(1 - 1/\nu^* - o(1))$-approximation, so in general, we know how to achieve a $\frac{1}{2}(1 - 1/\max\{\nu, \nu^*\} - o(1))$-approximation. For totally symmetric instances where $\nu = \nu^*$, we improve this to the optimal factor of $1 - 1/\nu$.

Proof: Since there is a unique center $c = \bar{x}$ for any $x \in B(\mathcal{M})$, this means this is also the symmetric optimum $F(c) = \max\{F(\bar{x}) : x \in B(\mathcal{M})\}$. Due to Lemma 5.2, c is a local optimum for the problem $\max\{F(x) : x \in B(\mathcal{M})\}$.

Because the fractional packing number of bases is ν, we have $c_i \leq 1/\nu$ for all i. Similarly, because the fractional packing number of dual bases (complements of bases) is ν^*, we have $1 - c_i \leq 1/\nu^*$. This means that $c \in [1 - 1/\nu^*, 1/\nu]$. Lemma 5.5 implies that

$$2F(c) \geq \left(1 - \frac{1}{\nu} + 1 - \frac{1}{\nu^*}\right) OPT.$$

■

Corollary 5.8. *Let $\max\{f(S) : S \in \mathcal{B}\}$ be an instance on a partition matroid where every base takes at least an α-fraction of each part, at most a $(1 - \alpha)$-fraction of each part, and the submodular function $f(S)$ is invariant under a group $\mathcal{G}$ where the orbit of each $i \in X_j$ is X_j. Then, the center point $c = \mathbf{E}_{\sigma \in \mathcal{G}}[\sigma(\mathbf{1}_B)]$ (equal for any $B \in \mathcal{B}$) satisfies $F(c) \geq \alpha \cdot OPT$.*

Proof: If the orbit of any element $i \in X_j$ is the entire set X_j, it also means that $\sigma(i)$ for a random $\sigma \in \mathcal{G}$ is uniformly distributed over X_j (by the transitive property of $\mathcal{G}$). Therefore, symmetrizing any fractional vector $x \in B(\mathcal{M})$ gives the same vector $\bar{x} = c$, where $c_i = k_j/|X_j|$ for $i \in X_j$. Also, our assumptions mean that the fractional packing number of bases is $1/(1 - \alpha)$, and the fractional packing number of dual bases is also $1/(1 - \alpha)$. Due to Lemma 5.7, the center c satisfies $F(c) \geq \alpha \cdot OPT$. ■

The hard instances for submodular maximization over matroid bases that we describe in Section 2 are exactly of this form (see the last paragraph of Section 2, with $\alpha = 1/k$). There is a unique symmetric solution, $x = (\alpha, \alpha, \ldots, \alpha, 1 - \alpha, 1 - \alpha, \ldots, 1 - \alpha)$. The fractional base packing number for these matroids is $\nu = 1/(1 - \alpha)$ and Theorem 1.6 implies that any $(\alpha + \epsilon) = (1 - 1/\nu + \epsilon)$-approximation for such matroids would require exponentially many value queries. Therefore, our approximation in this special case is optimal.

Acknowledgment

The author would like to thank Jon Lee and Maxim Sviridenko for helpful discussions.

References

[1] N. Alon and J. Spencer. The probabilistic method, 3rd edition, John Wiley & Sons, Hoboken, New Jersey, 2008.

[2] A. Ageev and M. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee, *J. of Combinatorial Optimization* 8 (2004), 307–328.

[3] G. Calinescu, C. Chekuri, M. Pál and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint, *Proc. of 12th IPCO* (2007), 182–196.

[4] G. Calinescu, C. Chekuri, M. Pál and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint, *to appear in SIAM J. on Computing, STOC 2008 special issue*.

[5] J. Edmonds. Matroids, submodular functions and certain polyhedra, *Combinatorial Structures and Their Applications* (1970), 69–87.

[6] J. Edmonds. Matroids and the greedy algorithm, *Math. Programming* 1, 127–136.

[7] U. Feige. A threshold of $\ln n$ for approximating Set Cover, *Journal of the ACM* 45 (1998), 634–652.

[8] U. Feige, V. Mirrokni and J. Vondrák. Maximizing non-monotone submodular functions, *Proc. of* 48^{th} *IEEE FOCS* (2007), 461-471.

[9] L. Fleischer, S. Fujishige and S. Iwata. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions, *Journal of the ACM* 48:4 (2001), 761–777.

[10] A. Frank. Matroids and submodular functions, *Annotated Biblographies in Combinatorial Optimization* (1997), 65–80.

[11] S. Fujishige. Canonical decompositions of symmetric submodular systems, *Discrete Applied Mathematics* 5 (1983), 175–190.

[12] S. Khot, R. Lipton, E. Markakis and A. Mehta. Inapproximability results for combinatorial auctions with submodular utility functions, in *Proc. of WINE 2005*.

[13] A. Kulik, H. Shachnai and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints, *Proc. of* 20^{th} *ACM-SIAM SODA* (2009), 545–554.

[14] J. Lee, V. Mirrokni, V. Nagarajan and M. Sviridenko. Maximizing non-monotone submodular functions under matroid and knapsack constraints, *Proc. of* 41^{st} *ACM STOC 2009*, 323–332.

[15] J. Lee, M. Sviridenko and J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties, *to appear in APPROX 2009*.

[16] L. Lovász. Submodular functions and convexity. A. Bachem et al., editors, *Mathematical Programmming: The State of the Art*, 235–257, 1983.

[17] V. Mirrokni, M. Schapira and J. Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions, *Proc. of EC 2008*.

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

[19] M. L. Fisher, G. L. Nemhauser and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions II, *Mathematical Programming Study* 8 (1978), 73–87.

[20] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function, *Math. Oper. Research*, 3(3):177–188, 1978.

[21] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory, Series B* 80 (2000), 346–355.

[22] A. Schrijver. Combinatorial optimization - polyhedra and efficiency. Springer, 2003.

[23] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint, *Operations Research Letters* 32 (2004), 41–43.

[24] J. Vondrák. Optimal approximation for the Submodular Welfare Problem in the value oracle model, *Proc. of* 40^{th} *ACM STOC 2008*, 67–74.

[25] J. Vondrák. Submodularity and curvature: the optimal algorithm, *to appear in RIMS Kokyuroku Bessatsu, Workshop on combinatorial optimization*, Kyoto 2008.

[26] L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem, *Combinatorica*, 2:385–393, 1982.

[27] L. Wolsey. Maximizing real-valued submodular functions: Primal and dual heuristics for location Problems, *Math. of Operations Research*, 7:410–425, 1982.

APPENDIX

1. Submodular functions and their extensions

In this appendix, we present a few basic facts concerning submodular functions $f(S)$ and their continuous extensions. By $f_A(S)$, we denote the marginal value of S w.r.t. A, $f_A(S) = f(A \cup S) - f(A)$. We also use $f_A(i) = f(A+i) - f(A)$. The notation $A+i$ is shorthand for $A \cup \{i\}$. Similarly, we write $A - i$ to denote $A \setminus \{i\}$.

Definition A.1. *The multilinear extension of a function* $f : 2^X \to \mathbb{R}$ *is a function* $F : [0,1]^X \to \mathbb{R}$ *where*

$$F(x) = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j).$$

Definition A.2. *The Lovász extension of a function* $f : 2^X \to \mathbb{R}$ *is a function* $\tilde{F} : [0,1]^X \to \mathbb{R}$ *such that*

$$\tilde{F}(x) = \sum_{i=0}^{n} (x_{\pi(i)} - x_{\pi(i+1)}) f(\{\pi(j) : 1 \leq j \leq i\})$$

where $\pi : [n] \to X$ *is a bijection such that* $x_{\pi(1)} \geq x_{\pi(2)} \geq \ldots \geq x_{\pi(n)}$*; we interpret* $x_{\pi(0)}$, $x_{\pi(n+1)}$ *as* $x_{\pi(0)} = 1, x_{\pi(n+1)} = 0$.

A useful way to view the multilinear extension is that we sample a random set $R(x)$, where each element i appears independently with probability x_i, and we take $F(x) = \mathbf{E}[f(R(x))]$. The Lovász extension can be viewed similarly, by sampling a random set in a correlated fashion.

Definition A.3. *For a vector* $x \in [0,1]^X$*, we define the "random threshold set"* $T(x)$ *by taking a uniformly random* $\lambda \in [0,1]$*, and setting*

$$T(x) = \{i \in X : x_i > \lambda\}.$$

Assuming that $x_1 \geq x_2 \geq \ldots \geq x_n$, $x_0 = 1$, $x_n = 0$, it is easy to see that the Lovász extension of $f(S)$ is equal to

$$\tilde{F}(x) = \sum_{i=0}^{n} (x_i - x_{i+1}) f([i]) = \mathbf{E}[f(T(x))].$$

It is known that the Lovász extension of a submodular function is always convex [16], which is not true for the multilinear extension. We prove that the Lovász extension is always upper-bounded by the multilinear extension; this lemma appears quite basic but it has not been published to our knowledge.

Lemma A.4. *Let* $F(x)$ *denote the multilinear extension and let* $\tilde{F}(x)$ *denote the Lovász extension of a submodular function* $f : 2^X \to \mathbb{R}$*. Then*

$$F(x) \geq \tilde{F}(x).$$

Proof: Let $R(x)$ be a random set where elements are sampled independently with probabilities x_i, and let $T(x)$ be the random threshold set. I.e., we want to prove $\mathbf{E}[f(R(x))] \geq \mathbf{E}[f(T(x))]$. We can assume WLOG that the elements are ordered so that $x_1 \geq x_2 \geq \ldots \geq x_n$. We also let $x_0 = 1$ and $x_{n+1} = 0$. Then,

$$\mathbf{E}[f(R(x))] = f(\emptyset) + \textstyle\sum_{k=1}^{n} \mathbf{E}[f_{R(x)\cap[k-1]}(R(x) \cap \{k\})]$$

and by submodularity,

$$\begin{aligned}\mathbf{E}[f(R(x))] &\geq f(\emptyset) + \sum_{k=1}^{n} \mathbf{E}[f_{[k-1]}(R(x) \cap \{k\})] \\ &= f(\emptyset) + \sum_{k=1}^{n} x_k (f([k]) - f([k-1])) \\ &= \sum_{k=0}^{n} (x_k - x_{k+1}) f([k]) \\ &= \mathbf{E}[f(T(x))].\end{aligned}$$

■

A refinement of this lemma says that we can also consider a partition of the ground set and apply an independent threshold set on each part. This gives a certain hybrid between the multilinear and Lovász extensions.

Lemma A.5. *For any partition $X = X_1 \cup X_2$,*

$$F(x) \geq \mathbf{E}[f((T_1(x) \cap X_1) \cup (T_2(x) \cap X_2))]$$

where $T_1(x)$ and $T_2(x)$ are two independently random threshold sets for x.

Proof: $F(x) = \mathbf{E}[f(R(x))]$ where $R(x)$ is sampled independently with probabilities x_i. Let's condition on $R(x) \cap X_2 = R_2$. This restricts the remaining randomness to X_1, and we get $f(R(x)) = f(R_2) + g(R(x))$, where $g(S) = f_{R_2}(S \cap X_1)$ is again submodular. By Lemma A.4, $\mathbf{E}[g(R(x))] \geq \mathbf{E}[g(T_1(x))]$ where $T_1(x)$ is a random threshold set. Hence,

$$\mathbf{E}[f(R(x)) \mid R(x) \cap X_2 = R_2] = f(R_2) + \mathbf{E}[g(R(x))]$$

$$\geq f(R_2) + \mathbf{E}[g(T_1(x))] = \mathbf{E}[f(R_2 \cup (T_1(x) \cap X_1))].$$

By randomizing $R_2 = R(x) \cap X_2$, we get

$$\mathbf{E}[f(R(x))] \geq \mathbf{E}[f((R(x) \cap X_2) \cup (T_1(x) \cap X_1))].$$

Repeating the same process, conditioning on $T_1(x)$ and applying Lemma A.4 to $R(x) \cap X_2$, we get

$$\mathbf{E}[f(R(x))] \geq \mathbf{E}[f((T_2(x) \cap X_2) \cup (T_1(x) \cap X_1))].$$

■

2. Pipage rounding for non-monotone submodular functions

The pipage rounding technique [2], [3], [4] starts with a point in the base polytope $y \in B(\mathcal{M})$ and produces an integral solution $S \in \mathcal{I}$ (in fact, a base) of expected value $\mathbf{E}[f(S)] \geq F(y)$. We recall the procedure here, in its randomized form [4].

Subroutine **HitConstraint**(y, i, j):
  Denote $\mathcal{A} = \{A \subseteq X : i \in A, j \notin A\}$;
  Find $\delta = \min_{A \in \mathcal{A}} (r_{\mathcal{M}}(A) - y(A))$
    and an optimal $A \in \mathcal{A}$;
  If $y_j < \delta$ then $\{\delta \leftarrow y_j,\ A \leftarrow \{j\}\}$;
  $y_i \leftarrow y_i + \delta,\ y_j \leftarrow y_j - \delta$;
  Return (y, A).

Algorithm **PipageRound**($(\mathcal{M}, y)$):
  While (y is not integral) do
    $T \leftarrow X$;
    While (T contains fractional variables) do
      Pick $i, j \in T$ fractional;
      $(y^+, A^+) \leftarrow \mathbf{HitConstraint}(y, i, j)$;
      $(y^-, A^-) \leftarrow \mathbf{HitConstraint}(y, j, i)$;
      $p \leftarrow ||y^+ - y|| / ||y^+ - y^-||$;
      With probability p, $\{y \leftarrow y^-, T \leftarrow T \cap A^-\}$;
        Else $\{y \leftarrow y^+, T \leftarrow T \cap A^+\}$;
    EndWhile
  EndWhile
  Output y.

The application in [3] was to monotone submodular functions, but as the authors mention, monotonicity is not used anywhere in the analysis. The technique as described in [4] yields the following.

Lemma A.6. *The pipage rounding technique, given a membership oracle for a matroid $\mathcal{M} = (X, \mathcal{I})$, a value oracle for a submodular function $f : 2^X \to \mathbb{R}_+$, and y in the base polytope $B(\mathcal{M})$, returns a random base B of value $\mathbf{E}[f(B)] \geq F(y)$.*

Monotonicity is used in [3] only to argue that a fractional solution can be assumed to lie in the base polytope without loss of generality. Therefore, if we are working with the base polytope (as in Section 4.2), we can use the pipage rounding technique without any modification.

If we are working with non-monotone submodular functions and the matroid polytope, we have to do some additional adjustments to make sure that we do not lose anything when rounding a fractional solution. We proceed as follows. The following procedure takes a point $x \in P(\mathcal{M})$ and while there is a fractional coordinate, it either pushes it to its maximum possible value, or makes it zero and removes it from the problem.

Algorithm **Adjust**($(\mathcal{M}, x)$):
  While (x is not in $B(\mathcal{M})$) do
    If (there is i and $\delta > 0$ such that $x + \delta \mathbf{e}_i \in P(\mathcal{M})$) do
      Let $x_{max} = x_i + \max\{\delta : x + \delta \mathbf{e}_i \in P(\mathcal{M})\}$;
      Let $p = x_i / x_{max}$;
      With probability p, $\{x_i \leftarrow x_{max}\}$;
        Else $\{x_i \leftarrow 0\}$;
    EndIf
    If (there is i such that $x_i = 0$) do
      Delete i from $\mathcal{M}$ and remove the i-coordinate from x.
  EndWhile
  Output $(\mathcal{M}, x)$.

Lemma A.7. *Given $x \in P(\mathcal{M})$ and a submodular function*

$f(S)$ with its extension $F(x)$, the procedure **Adjust***($(\mathcal{M}, x)$) yields a restricted matroid $\mathcal{M}'$ and a point $y \in B(\mathcal{M}')$ such that* $\mathbf{E}[F(y)] \geq F(x)$.

Proof: If $x \in P(\mathcal{M})$, there is always a point $z \in B(\mathcal{M})$ dominating x in every coordinate (see, e.g., Corollary 40.2h in [22]). Therefore, if $x \notin B(\mathcal{M})$, there is a coordinate which can be increased while staying in $P(\mathcal{M})$. In each step, when we choose randomly between increasing and decreasing x_i, the objective function is linear and the expectation of $F(x)$ is preserved. Hence, the process forms a martingale. At the end, $\mathbf{E}[F(y)]$ is equal to the initial value $F(x)$.

As long as we increase variables, each variable is increased to its maximum value and cannot be increased twice. Therefore, after at most n steps we either reach a point in $B(\mathcal{M})$ or make some variable x_i equal to 0. A variable equal to 0 is removed, which can be repeated at most n times. Hence, the process terminates in $O(n^2)$ time. ∎

For a given $x \in P(\mathcal{M})$, we run $(\mathcal{M}', y) :=$**Adjust**$(\mathcal{M}, x)$, followed by **PipageRound**$((\mathcal{M}', y))$. The outcome is a base in the restricted matroid where some elements have been deleted, i.e. an independent set in the original matroid. We call this the *extended pipage rounding* procedure. Lemma A.7 together with Lemma A.6 gives the following.

Lemma A.8. *The extended pipage rounding procedure, given a membership oracle for a matroid $\mathcal{M} = (X, \mathcal{I})$, a value oracle for a submodular function $f : 2^X \to \mathbb{R}_+$, and x in the matroid polytope $P(\mathcal{M})$, yields a random independent set $S \in \mathcal{I}$ of value* $\mathbf{E}[f(S)] \geq F(x)$.

3. *Solutions of non-trivial value*

In this section, we prove that solutions of non-trivial value always exist for the problems of maximizing a non-monotone submodular function subject to a matroid independence or matroid base constraint. We can assume that our matroid does not contain any loops (which can be removed beforehand), and in the case of a matroid base constraint no co-loops either (since co-loops participate in every solution and can be contracted). The bounds that we prove here are needed to ensure that our sampling errors can be made negligible compared to the value of the optimum.

Lemma A.9. *Let $\mathcal{M} = (X, \mathcal{I})$ be a matroid without loops (elements which are never in an independent set), $|X| = n$, and let $f : 2^X \to \mathbb{R}_+$ be a (non-monotone) submodular function such that $\max_{S \subseteq X} f(S) = M$. Let $OPT = \max\{f(I) : I \in \mathcal{I}\}$. Then*

$$OPT \geq \frac{1}{n} M.$$

Proof: We consider only solutions of size $|I| \leq 1$, which are independent because our matroid does not contain any loops. Let $f(I^*) = \max\{f(I) : |I| \leq 1\}$. By submodularity and nonnegativity of f, the value of any non-empty set can be estimated as

$$f(S) \leq \sum_{j \in S} f(\{j\}) \leq n\, f(I^*).$$

By definition, we also have $f(\emptyset) \leq f(I^*)$. Therefore, $M = \max f(S) \leq n\, f(I^*)$. ∎

Lemma A.10. *Let $\mathcal{M} = (X, \mathcal{I})$ be a matroid without loops (elements which are never in a base) and coloops (elements which are in every base), $|X| = n$ and let $\mathcal{B}$ denote the bases of $\mathcal{M}$. Let $f : 2^X \to \mathbb{R}_+$ be a (non-monotone) submodular function such that $\max_{S \subseteq X} f(S) = M$. Let $OPT = \max\{f(B) : B \in \mathcal{B}\}$. Then*

$$OPT \geq \frac{1}{n^2} M.$$

Proof: Let us find $B = \{b_1, \ldots, b_k\}$ greedily, by including in each step an element which has maximum possible marginal value $f_{\{b_1,\ldots,b_\ell\}}(b_{\ell+1})$ among all elements that preserve independence. We claim that $f(B) \geq \frac{1}{n^2} M$.

The first element b_1 is the element of maximum value $\max_{i \in X} f(\{i\})$ (because there are no loops, all elements are eligible). Let $M = f(S^*)$. By submodularity,

$$f(S^*) \leq \sum_{i \in S^*} f(\{i\}) \leq n f(\{b_1\}).$$

Therefore, $f(\{b_1\}) \geq \frac{1}{n} M$. However, additional elements can decrease this value. Consider the last element b_k that we added to B and assume that $f(B) < \frac{1}{n^2} M$. Due to our greedy procedure and submodularity, the marginal values of successive elements keep decreasing, and therefore

$$f_{B \setminus \{b_k\}}(b_k) \leq \frac{1}{n-1}(f(B) - f(\{b_1\})) < -\frac{1}{n^2} M.$$

Since there are no coloops in $\mathcal{M}$, adding b_k was not our only choice - otherwise, we would have $\text{rank}(X \setminus \{b_k\}) < \text{rank}(X)$ which means exactly that b_k is a coloop. So there is another element b'_k that would form a base $\{b_1, \ldots, b_{k-1}, b'_k\}$. The reason we did not select b'_k was that $f_{B \setminus \{b_k\}}(b'_k) \leq f_{B \setminus \{b_k\}}(b_k) < -\frac{1}{n^2} M$. By submodularity, if we add both elements, we obtain a set $\tilde{B} = \{b_1, \ldots, b_k, b'_k\}$ such that

$$f(\tilde{B}) = f(B) + f_B(b'_k) \leq f(B) + f_{B \setminus \{b_k\}}(b'_k) < \frac{1}{n^2} M - \frac{1}{n^2} M = 0$$

which is a contradiction with the nonnegativity of f. ∎

We remark that this bound is tight up to a constant factor. Consider a complete bipartite directed graph on $X = X_1 \cup X_2$, $|X_1| = |X_2| = n/2$. The submodular function $f(S)$ is the directed cut function, expressing the number of arcs from S to $\bar{S}$. The matroid $\mathcal{M}$ has bases that contain exactly one element from X_1 and $n-1$ elements from X_2. It is easy to see that any base B has value $f(B) = 1$, while $f(X_1) = n^2/4$.