SIAM J. DISCRETE MATH.
Vol. 0, No. 0, pp. 000–000

MAXIMIZING NONMONOTONE SUBMODULAR FUNCTIONS UNDER MATROID OR KNAPSACK CONSTRAINTS*

JON LEE†, VAHAB S. MIRROKNI‡,
VISWANATH NAGARAJAN†, AND MAXIM SVIRIDENKO†

Abstract. Submodular function maximization is a central problem in combinatorial optimization, generalizing many important problems including Max Cut in directed/undirected graphs and in hypergraphs, certain constraint satisfaction problems, maximum entropy sampling, and maximum facility location problems. Unlike submodular minimization, submodular maximization is NP-hard. In this paper, we give the first constant-factor approximation algorithm for maximizing any nonnegative submodular function subject to multiple matroid or knapsack constraints. We emphasize that our results are for *nonmonotone* submodular functions. In particular, for any constant k, we present a $(\frac{1}{k+2+\frac{1}{k}+})$-approximation for the submodular maximization problem under k matroid constraints, and a $(\frac{1}{5}-\epsilon)$-approximation algorithm for this problem subject to k knapsack constraints ($\epsilon > 0$ is any constant). We improve the approximation guarantee of our algorithm to $\frac{1}{k+1+\frac{1}{k-1}+\epsilon}$ for $k \geq 2$ partition matroid constraints. This idea also gives a $(\frac{1}{k+\epsilon})$-approximation for maximizing a *monotone* submodular function subject to $k \geq 2$ partition matroids, which is an improvement over the previously best known guarantee of $\frac{1}{k+1}$.

Key words.

AMS subject classifications.

DOI. 10.1137/090750020

1. Introduction.

In this paper, we consider the problem of maximizing a *nonnegative submodular function f, defined on a ground set V, subject to matroid constraints or knapsack constraints.* A function $f : 2^V \to \mathbb{R}$ is *submodular* if for all $S, T \subseteq V$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$. Throughout, we assume that our submodular function f is given by a *value oracle*; i.e., for a given set $S \subseteq V$, an algorithm can query an oracle to find its value $f(S)$. Furthermore, all submodular functions that we deal with are assumed to be nonnegative. We also denote the ground set $V = [n] = \{1, 2, \ldots, n\}$.

We emphasize that our focus is on submodular functions that are *not required to be monotone* (i.e., we do *not* require that $f(X) \leq f(Y)$ for $X \subseteq Y \subseteq V$). Nonmonotone submodular functions appear in several places including cut functions in weighted directed or undirected graphs or even hypergraphs, maximum facility location, maximum entropy sampling, and certain constraint satisfaction problems.

Given a weight vector w for the ground set V and a knapsack of capacity C, the associated *knapsack constraint* is that the sum of weights of elements in the solution S should not exceed the capacity C, i.e, $\sum_{j \in S} w_j \leq C$. In our usage, we consider k knapsack constraints defined by weight vectors w^i and capacities C_i for $i = 1, \ldots, k$.

We assume some familiarity with matroids [43] and associated algorithmics [48]. Briefly, for a matroid $\mathcal{M}$, we denote the ground set of $\mathcal{M}$ by $\mathcal{E}(\mathcal{M})$, its set of

*Received by the editors February 17, 2009; accepted for publication (in revised form) September 15, 2009; published electronically DATE.
http://www.siam.org/journals/sidma/x-x/75002.html
†Mathematical Sciences, IBM T.J. Watson Research Center, Parktown Heights, NY 10598 (jonlee@us.ibm.com, viswanath@us.ibm.com, sviri@us.ibm.com).
‡Google Research, New York, NY 10011 (mirrokni@gmail.com).

independent sets by $\mathcal{I}(\mathcal{M})$, and its set of bases by $\mathcal{B}(\mathcal{M})$. For a given matroid $\mathcal{M}$, the associated *matroid constraint* is $S \in \mathcal{I}(\mathcal{M})$ and the associated *matroid base constraint* is $S \in \mathcal{B}(\mathcal{M})$. As is standard, $\mathcal{M}$ is a *uniform matroid* of rank r if $\mathcal{I}(\mathcal{M}) := \{X \subseteq \mathcal{E}(\mathcal{M}) \ : \ |X| \leq r\}$. A *partition matroid* is the direct sum of uniform matroids. Note that uniform matroid constraints are equivalent to cardinality constraints, i.e, $|S| \leq k$. In our usage, we deal with k matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$ on the common ground set $V := \mathcal{E}(\mathcal{M}_1) = \cdots = \mathcal{E}(\mathcal{M}_k)$ (which is also the ground set of our submodular function f), and we let $\mathcal{I}_i := \mathcal{I}(\mathcal{M}_i)$ for $i = 1, \ldots, k$.

Background. Optimizing submodular functions is a central subject in operations research and combinatorial optimization [38]. It appears in many important optimization contexts including cuts in graphs [19, 44, 26], rank functions of matroids [12, 16], set covering problems [13], plant location problems [9, 10, 11, 2], certain satisfiability problems [25, 14], and maximum entropy sampling [32, 33]. Other than many heuristics that have been developed for optimizing these functions [20, 21, 27, 46, 31], many exact and constant-factor approximation algorithms are also known for this problem [41, 42, 47, 26, 15, 52, 18]. In some settings such as set covering or matroid optimization, the relevant submodular functions are monotone. Here we are more interested in the general case where $f(S)$ is not necessarily monotone.

Unlike submodular minimization [47, 26], submodular function maximization is NP-hard as it generalizes many NP-hard problems such as Max Cut [19, 14] and maximum facility location [9, 10, 2]. Other than generalizing combinatorial optimization problems like Max Cut [19], Max Directed Cut [4, 22], hypergraph cut problems, maximum facility location [2, 9, 10], and certain restricted satisfiability problems [25, 14], maximizing nonmonotone submodular functions has applications in a variety of problems, e.g., computing the core value of supermodular games [49] and optimal marketing for revenue maximization over social networks [23]. As an example, we describe one important application in the statistical design of experiments. Let A be the n-by-n covariance matrix of a set of Gaussian random variables indexed by $[n]$. For $S \subseteq [n]$, let $A[S]$ denote the principal submatrix of A indexed by S. It is well known that the entropy[1] of the random variables indexed by S is

$$f(S) = \frac{1 + \ln(2\pi)}{2}|S| + \frac{1}{2}\ln\det A[S] \ .$$

Certainly $|S|$ is nonnegative, monotone, and (sub)modular on $[n]$. It is also well known that $\ln\det A[S]$ is submodular on $[n]$, but $\ln\det A[S]$ is not even approximately monotone (see [30, section 8.2]). For example, for

$$A = \begin{pmatrix} \delta & \sqrt{\delta - 1} \\ \sqrt{\delta - 1} & 1 \end{pmatrix},$$

with $\delta > 1$, it is clear that $\ln\det A[\{1,2\}] = 0$, while $\ln\det A[\{1\}] = \ln(\delta)$ can be made arbitrarily large by taking δ large. So the entropy $f(S)$ is submodular but not generally monotone. The *maximum entropy sampling problem*, introduced in [50], is to maximize $f(S)$ over subsets $S \subseteq [n]$ having cardinality s fixed. So the maximum entropy sampling problem is precisely one of maximizing a nonmonotone submodular function subject to a cardinality constraint. Of course a cardinality constraint is just a matroid base constraint for a uniform matroid. The maximum entropy sampling problem has mostly been studied from a computational point of view (often in

[1] Sometimes also referred to as *differential entropy* or *continuous entropy*.

the context of locating environmental monitoring stations), focusing on calculating optimal solutions for moderate-sized instances (say, $n < 200$) using mathematical programming methodologies (e.g., see [32, 33, 34, 29, 6, 5]), and our results provide the first set of algorithms with provable constant-factor approximation guarantee (for cases in which the entropy is nonnegative).

Recently, a $\frac{2}{5}$-approximation was developed for maximizing nonnegative nonmonotone submodular functions without any side constraints [15]. This algorithm also provides a tight $\frac{1}{2}$-approximation algorithm for maximizing a symmetric[2] submodular function [15]. However, the algorithms developed in [15] for nonmonotone submodular maximization do not handle any extra constraints.

For the problem of maximizing a monotone submodular function subject to a matroid or multiple knapsack constraints, tight $\left(1 - \frac{1}{e}\right)$-approximations are known [41, 7, 53, 51, 28]. Maximizing monotone submodular functions over k matroid constraints was considered in [42], where a $(\frac{1}{k+1})$-approximation was obtained. This bound is currently the best known ratio, even in the special case of partition matroid constraints. However, none of these results generalize to nonmonotone submodular functions.

Better results are known either for specific submodular functions or for special classes of matroids. A $\frac{1}{k}$-approximation algorithm using local search was designed in [45] for the problem of maximizing a linear function subject to k matroid constraints. Constant factor approximation algorithms are known for the problem of directed cut [1] or hypergraph cut [3] maximization subject to a uniform matroid (i.e., cardinality) constraint.

Hardness of approximation results are known even for the special case of maximizing a linear function subject to k partition matroid constraints. The best known lower bound is an $\Omega(\frac{k}{\log k})$ hardness of approximation [24]. Moreover, for the unconstrained maximization of nonmonotone submodular functions, it has been shown that achieving a factor better than $\frac{1}{2}$ cannot be done using a subexponential number of value queries [15].

Our results. In this paper, we give the first constant-factor approximation algorithms for maximizing a nonmonotone submodular function subject to multiple matroid constraints or multiple knapsack constraints. More specifically, we give the following new results (below $\epsilon > 0$ is any constant).

(1) For every constant $k \geq 1$, we present a $(\frac{1}{k+2+\frac{1}{k}+\epsilon})$-approximation algorithm for maximizing any nonnegative submodular function subject to k matroid constraints (see section 2). This implies a $(\frac{1}{4+\epsilon})$-approximation algorithm for maximizing nonmonotone submodular functions subject to a single matroid constraint. Moreover, this algorithm is a $(\frac{1}{k+2+\epsilon})$-approximation in the case of *symmetric* submodular functions. This algorithm involves a natural local search procedure that is iteratively executed $k+1$ times. Asymptotically, this result is nearly the best possible because there is an $\Omega(\frac{k}{\log k})$ hardness of approximation, even in the monotone case [24].

(2) For every constant $k \geq 1$, we present a $\left(\frac{1}{5} - \epsilon\right)$-approximation algorithm for maximizing any nonnegative submodular function subject to a k-dimensional knapsack constraint (see section 3). To achieve the approximation guarantee, we first give a $\left(\frac{1}{4} - \epsilon\right)$-approximation algorithm for a fractional relaxation

[2] The function $f : 2^V \to \mathbb{R}$ is symmetric if for all $S \subseteq V$, $f(S) = f(V \setminus S)$. For example, cut functions in undirected graphs are well known examples of symmetric (nonmonotone) submodular functions.

(similar to the one used in [53]). This is again based on a local search procedure that is iterated twice. We then use a simple randomized rounding technique to convert a fractional solution to an integral one. A similar approach was recently used in [28] for maximizing a monotone submodular function over multiple knapsack constraints. However, the algorithm of [28] for the fractional relaxation uses the "continuous greedy" algorithm of Vondrák [53] which requires a monotone function; moreover, even their rounding method is not directly applicable to nonmonotone submodular functions.

(3) For submodular maximization under $k \geq 2$ *partition matroid* constraints, we obtain improved approximation guarantees (see section 4). We give a $(\frac{1}{k+1+\frac{1}{k-1}+\epsilon})$-approximation algorithm for maximizing nonmonotone submodular functions subject to k partition matroids. Moreover, our idea gives a $(\frac{1}{k+\epsilon})$-approximation algorithm for maximizing a monotone submodular function subject to $k \geq 2$ partition matroid constraints. This is an improvement over the previously best known bound of $\frac{1}{k+1}$ from [42].

(4) Finally, we study submodular maximization subject to a matroid *base* constraint in section 5. We give a $\left(\frac{1}{3} - \epsilon\right)$-approximation in the case of symmetric submodular functions. Our result for general submodular functions holds only for special matroids: We obtain a $(\frac{1}{6} - \epsilon)$-approximation when the matroid contains two disjoint bases. In particular, this implies a $\left(\frac{1}{6} - \epsilon\right)$-approximation for the problem of maximizing any nonnegative submodular function subject to an exact cardinality constraint. Previously, only special cases of directed cut [1] or hypergraph cut [3] subject to an exact cardinality constraint were considered.

All our algorithms run in time $n^{O(k)}$, where k is the number of matroid or knapsack constraints.

Our main technique for the above results is local search. Our local search algorithms are different from the previously used variant of local search for unconstrained maximization of a nonnegative submodular function [15], or the local search algorithms used for Max Directed Cut [4, 22]. In the design of our algorithms, we also use structural properties of matroids, a fractional relaxation of submodular functions, and a randomized rounding technique.

There are a few new results worth mentioning that appeared after publication of the preliminary versions of this paper [35, 36]. Lee, Sviridenko, and Vondrák [37] considered the more general version of the local search algorithm that adds up to p and deletes up to kp elements from the current solution for the problem under $k \geq 2$ general matroid constraints, where $p \geq 1$ is an arbitrary fixed constant. This result is analogous to the result in the present paper for $k \geq 2$ partition matroid constraints. However, the difficulty in the general case is that [37] needed to establish new matroid exchange properties to carry out the analysis. Another recent development is the paper of Vondrák [54] where (among other results) he shows that there is no polynomial time constant factor approximation algorithm for the problem of submodular maximization under a matroid base constraint.

2. Submodular maximization subject to k matroid constraints. In this section, we give an approximation algorithm for submodular maximization subject to k matroid constraints. The problem is as follows: Let f be a *nonnegative* submodular function defined on ground set V. Let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be k arbitrary matroids on the common ground set V. For each matroid $\mathcal{M}_j$ (with $j \in [k]$) we denote the set of its independent sets by $\mathcal{I}_j$. We consider the following problem:

$$\max\left\{f(S) \;:\; S \in \cap_{j=1}^{k}\mathcal{I}_j\right\}. \tag{2.1}$$

We give an approximation algorithm for this problem using value queries to f that runs in time $n^{O(k)}$. The starting point is the following local search algorithm. Starting with $S = \emptyset$, repeatedly perform one of the following local improvements:

- *Delete operation.* If $e \in S$ such that $f(S \setminus \{e\}) > f(S)$, then $S \leftarrow S \setminus \{e\}$.
- *Exchange operation.* If $d \in V \setminus S$ and $e_i \in S \cup \{\emptyset\}$ (for $1 \le i \le k$) are such that $(S \setminus \{e_i\}) \cup \{d\} \in \mathcal{I}_i$ for all $i \in [k]$ and $f((S \setminus \{e_1, \dots, e_k\}) \cup \{d\}) > f(S)$, then $S \leftarrow (S \setminus \{e_1, \dots, e_k\}) \cup \{d\}$.

For notational convenience, we regard $\emptyset$ as a dummy element not in the ground set $[n]$; in particular, the exchange operation allows dropping *up to* k elements.

When dealing with a single matroid constraint ($k = 1$), the local operations correspond to the following: *delete* an element, *add* an element (i.e., an exchange when no element is dropped), and *swap* a pair of elements (i.e., an element from outside the current set is exchanged with an element from the set). With $k \ge 2$ matroid constraints, we permit more general exchange operations, involving adding one element and dropping at most k elements.

Note that the size of any local neighborhood is at most n^{k+1}, which implies that each local step can be performed in polynomial time for a constant k. Let S denote a locally optimal solution. Next, we prove a key lemma for this local search algorithm, which is used in analyzing our algorithm. Before presenting the lemma, we state a useful exchange property of matroids (see [48]). Intuitively, this property states that for any two independent sets I and J, we can add any element of J to the set I and remove at most one element from I while keeping the set independent. Moreover, each element of I is allowed to be displaced by at most one element of J.

THEOREM 2.1. *Let $\mathcal{M}$ be a matroid and $I, J \in \mathcal{I}(\mathcal{M})$ be two independent sets. Then there is a mapping $\pi : J \setminus I \to (I \setminus J) \cup \{\emptyset\}$ such that*

1. $(I \setminus \pi(b)) \cup \{b\} \in \mathcal{I}(\mathcal{M})$ *for all* $b \in J \setminus I$*;*
2. $|\pi^{-1}(e)| \le 1$ *for all* $e \in I \setminus J$.

Proof. We outline the proof for completeness. We proceed by induction on $t = |J \setminus I|$. If $t = 0$, there is nothing to prove; so assume $t \ge 1$. Suppose there is an element $b \in J \setminus I$ with $I \cup \{b\} \in \mathcal{I}(\mathcal{M})$. In this case we apply induction on I and $J' = J \setminus \{b\}$ (where $|J' \setminus I| = t - 1 < t$). Since $I \setminus J' = I \setminus J$, we obtain a map $\pi' : J' \setminus I \to (I \setminus J) \cup \{\emptyset\}$ satisfying the two conditions. The desired map π for $\langle I, J\rangle$ is then $\pi(b) = \emptyset$ and $\pi(b') = \pi'(b')$ for all $b' \in J \setminus I \setminus \{b\} = J' \setminus I$.

Now we may assume that I is a maximal independent set in $I \cup J$. Let $\mathcal{M}' \subseteq \mathcal{M}$ denote the matroid $\mathcal{M}$ restricted to $I \cup J$; so I is a base in $\mathcal{M}'$. We augment J to some base $\tilde{J} \supseteq J$ in $\mathcal{M}'$ (since any maximal independent set in $\mathcal{M}'$ is a base). Thus we have two bases I and $\tilde{J}$ in $\mathcal{M}'$. Theorem 39.12 from [48] implies the existence of elements $b \in \tilde{J} \setminus I$ and $e \in I \setminus \tilde{J}$ such that both $(\tilde{J} \setminus b) \cup \{e\}$ and $(I \setminus e) \cup \{b\}$ are bases in $\mathcal{M}'$. Note that $J' := (J \setminus \{b\}) \cup \{e\} \subseteq (\tilde{J} \setminus \{b\}) \cup \{e\} \in \mathcal{I}(\mathcal{M})$; also $I \setminus J' = (I \setminus J) \setminus \{e\}$ and $J' \setminus I = (J \setminus I) \setminus \{b\}$. By induction on I and J' (since $|J' \setminus I| = t - 1 < t$) we obtain map $\pi' : J' \setminus I \to I \setminus J'$ satisfying the two conditions. The map π for $\langle I, J\rangle$ is then $\pi(b) = e$ and $\pi(b') = \pi'(b')$ for all $b' \in (J \setminus I) \setminus \{b\} = J' \setminus I$. The first condition on π is satisfied by induction (for elements $(J \setminus I) \setminus \{b\}$) and since $(I \setminus e) \cup \{b\} \in \mathcal{I}(\mathcal{M})$ (see above). The second condition on π is satisfied by induction and the fact that $e \notin I \setminus J'$. ∎

LEMMA 2.2. *For a locally optimal solution S and any $C \in \cap_{j=1}^{k}\mathcal{I}_j$, $(k+1) \cdot f(S) \ge f(S \cup C) + k \cdot f(S \cap C)$. Additionally for $k = 1$, if $S \in \mathcal{I}_1$ is any locally* *optimal solution under only the swap operation and* $C \in \mathcal{I}_1$ *with* $|S| = |C|$*, then* $2 \cdot f(S) \geq f(S \cup C) + f(S \cap C)$.

Proof. The following proof is due to Vondrák [55]. Our original proof [35] was more complicated—we thank Vondrák for letting us present this simplified proof.

For each matroid $\mathcal{M}_j$ ($j \in [k]$), because both $C, S \in \mathcal{I}_j$ are independent sets, Theorem 2.1 implies a mapping $\pi_j : C \setminus S \to (S \setminus C) \cup \{\emptyset\}$ such that

1. $(S \setminus \pi_j(b)) \cup \{b\} \in \mathcal{I}_j$ for all $b \in C \setminus S$;
2. $|\pi_j^{-1}(e)| \leq 1$ for all $e \in S \setminus C$.

When $k = 1$ and $|S| = |C|$, Corollary 39.12(a) from [48] implies the stronger condition that $\pi_1 : C \setminus S \to S \setminus C$ is in fact a *bijection.*

For each $b \in C \setminus S$, let $A_b = \{\pi_1(b), \ldots, \pi_k(b)\}$. Note that $(S \setminus A_b) \cup \{b\} \in \cap_{j=1}^k \mathcal{I}_j$ for all $b \in C \setminus S$. Hence $(S \setminus A_b) \cup \{b\}$ is in the local neighborhood of S, and by local optimality under exchanges

$$f(S) \geq f\left((S \setminus A_b) \cup \{b\}\right) \qquad \forall b \in C \setminus S. \tag{2.2}$$

In the case $k = 1$ with $|S| = |C|$, these are only *swap* operations (because π_1 is a bijection here).

By the property of mappings $\{\pi_j\}_{j=1}^k$, each element $i \in S \setminus C$ is contained in $n_i \leq k$ of the sets $\{A_b \mid b \in C \setminus S\}$; and elements of $S \cap C$ are contained in none of these sets. So the following inequalities are implied by local optimality of S under deletions:

$$(k - n_i) \cdot f(S) \geq (k - n_i) \cdot f(S \setminus \{i\}) \qquad \forall i \in S \setminus C. \tag{2.3}$$

Note that these inequalities are not required when $k = 1$ and $|S| = |C|$ (then $n_i = k$ for all $i \in S \setminus C$).

For any $b \in C \setminus S$, we have the following (below, the first inequality is submodularity and the second is from (2.2)):

$$f(S \cup \{b\}) - f(S) \leq f\left((S \setminus A_b) \cup \{b\}\right) - f\left(S \setminus A_b\right) \leq f(S) - f\left(S \setminus A_b\right).$$

Adding this inequality over all $b \in C \setminus S$ and using submodularity,

$$f(S \cup C) - f(S) \leq \sum_{b \in C \setminus S} \left[f(S \cup \{b\}) - f(S)\right] \leq \sum_{b \in C \setminus S} \left[f(S) - f\left(S \setminus A_b\right)\right].$$

Adding to this the inequalities (2.3), i.e., $0 \leq (k - n_i) \cdot [f(S) - f(S \setminus \{i\})]$ for all $i \in S \setminus C$,

$$\begin{aligned} f(S \cup C) - f(S) &\leq \sum_{b \in C \setminus S} \left[f(S) - f\left(S \setminus A_b\right)\right] + \sum_{i \in S \setminus C} (k - n_i) \cdot \left[f(S) - f(S \setminus \{i\})\right] \\ &= \sum_{l=1}^{\lambda} \left[f(S) - f\left(S \setminus T_l\right)\right], \qquad (2.4) \end{aligned}$$

where $\lambda = |C \setminus S| + \sum_{i \in S \setminus C} (k - n_i)$ and $\{T_l\}_{l=1}^{\lambda}$ is some collection of subsets of $S \setminus C$ such that each $i \in S \setminus C$ appears in *exactly* k of these subsets. We simplify the expression (2.4) using the following claim.

Claim 2.3. *Let* $f : 2^V \to \mathbb{R}_+$ *be any submodular function and* $S' \subseteq S \subseteq V$*. Let* $\{T_l\}_{l=1}^{\lambda}$ *be a collection of subsets of* $S \setminus S'$ *such that each element of* $S \setminus S'$ *appears in* *exactly k of these subsets. Then*

$$\sum_{l=1}^{\lambda} [f(S) - f(S \setminus T_l)] \leq k \cdot (f(S) - f(S')) .$$

Proof. Let $s = |S|$ and $|S'| = c$; number the elements of S as $\{1, 2, \ldots, s\} = [s]$ such that $S' = \{1, 2, \ldots, c\} = [c]$. Then for any $T \subseteq S \setminus S'$, by submodularity, $f(S) - f(S \setminus T) \leq \sum_{p \in T} [f([p]) - f([p-1])]$. Using this we obtain the following:

$$\begin{aligned} \sum_{l=1}^{\lambda} [f(S) - f(S \setminus T_l)] &\leq \sum_{l=1}^{\lambda} \sum_{p \in T_l} [f([p]) - f([p-1])] \\ &= k \sum_{i=c+1}^{s} [f([i]) - f([i-1])] = k \cdot (f(S) - f(S')) . \end{aligned}$$

The second equality follows from $S \setminus C = \{c+1, \ldots, s\}$ and the fact that each element of $S \setminus C$ appears in exactly k of the sets $\{T_l\}_{l=1}^{\lambda}$. The last equality is due to a telescoping summation. ∎

Setting $S' = S \cap C$ in Claim 2.3 to simplify expression (2.4), we obtain $(k+1) \cdot f(S) \geq f(S \cup C) + k \cdot f(S \cap C)$.

Observe that when $k = 1$ and $|S| = |C|$, we used only the inequalities (2.2) from the local search, which are only swap operations. Hence in this case, the statement also holds for any solution S that is locally optimal under only swap operations. In the general case, we use both inequalities (2.2) (from exchange operations) and inequalities (2.3) (from deletion operations). ∎

A simple consequence of Lemma 2.2 implies bounds analogous to known approximation factors [42, 45] in the cases when the submodular function f has additional structure.

COROLLARY 2.4. *For a locally optimal solution S and any $C \in \cap_{j=1}^{k} \mathcal{I}_j$ the following inequalities hold:*

1. $f(S) \geq f(C)/(k+1)$ *if function f is monotone;*
2. $f(S) \geq f(C)/k$ *if function f is linear.*

The local search algorithm defined above could run for an exponential amount of time until it reaches a locally optimal solution. To ensure polynomial runtime, we follow the standard approach of an approximate local search under a suitable (small) parameter $\epsilon > 0$ as described in Figure 2.1. The following Lemma 2.5 is a simple extension of Lemma 2.2 for approximate local optimum.

LEMMA 2.5. *For an approximately locally optimal solution S (in Procedure B) and any $C \in \cap_{j=1}^{k} \mathcal{I}_j$, $(1+\epsilon)(k+1) \cdot f(S) \geq f(S \cup C) + k \cdot f(S \cap C)$, where $\epsilon > 0$*

APPROXIMATE LOCAL SEARCH PROCEDURE B.

Input: Ground set X of elements and value oracle access to submodular function f.

1. Set $v \leftarrow \arg\max\{f(u) \mid u \in X\}$ and $S \leftarrow \{v\}$.
2. While one of the following local operations applies, update S accordingly.
   - **Delete operation on S.** If $e \in S$ such that $f(S \setminus \{e\}) \geq (1 + \frac{\epsilon}{n^4}) f(S)$, then $S \leftarrow S \setminus \{e\}$.
   - **Exchange operation on S.** If $d \in X \setminus S$ and $e_i \in S \cup \{\emptyset\}$ (for $1 \leq i \leq k$) are such that $(S \setminus \{e_i\}) \cup \{d\} \in \mathcal{I}_i$ for all $i \in [k]$ and $f((S \setminus \{e_1, \ldots, e_k\}) \cup \{d\}) \geq (1 + \frac{\epsilon}{n^4}) f(S)$, then $S \leftarrow (S \setminus \{e_1, \ldots, e_k\}) \cup \{d\}$.

FIG. 2.1. *The approximate local search procedure.*

ALGORITHM A.

1. Set $V_1 = V$.
2. For $i = 1, \ldots, k+1$, do:
   (a) Apply the approximate local search Procedure B on the ground set V_i to obtain a solution $S_i \subseteq V_i$ corresponding to the problem:
   $$\max\{f(S) \; : \; S \in \cap_{j=1}^k \mathcal{I}_j, \; S \subseteq V_i\}. \tag{2.5}$$
   (b) Set $V_{i+1} = V_i \setminus S_i$.
3. Return the solution corresponding to $\max\{f(S_1), \ldots, f(S_{k+1})\}$.

FIG. 2.2. *Approximation algorithm for submodular maximization under k matroid constraints.*

is the parameter used in the Procedure B (see Figure 2.1). Additionally for $k = 1$, if $S \in \mathcal{I}_1$ is any locally optimal solution under only the swap operation, and $C \in \mathcal{I}_1$ with $|S| = |C|$, then $2(1+\epsilon) \cdot f(S) \geq f(S \cup C) + f(S \cap C)$.

Proof. The proof of this lemma is almost identical to the proof of Lemma 2.2; the only difference is that the left-hand sides of inequalities (2.2) and inequalities (2.3) are multiplied by $1 + \frac{\epsilon}{n^4}$. Therefore, after following the steps in Lemma 2.2, we obtain the following inequality:

$$\left(k + 1 + \frac{\epsilon}{n^4}\lambda\right) \cdot f(S) \geq f(S \cup C) + k \cdot f(S \cap C).$$

Since $\lambda \leq (k+1)n$ (see Lemma 2.2) and we may assume that $n^4 >> (k+1)n$, we obtain the lemma. ∎

We now present the main algorithm (Figure 2.2) for submodular maximization over matroid constraints. This performs the approximate local search Procedure B iteratively $k+1$ times and outputs the best solution found.

THEOREM 2.6. *Algorithm A in Figure 2.2 is a $\left(\frac{1}{(1+\epsilon)(k+2+\frac{1}{k})}\right)$-approximation algorithm for maximizing a nonnegative submodular function subject to any k matroid constraints, running in time $n^{O(k)}$.*

Proof. Bounding the running time of Algorithm A is easy. The parameter $\epsilon > 0$ in Procedure B is any value such that $\frac{1}{\epsilon}$ is at most a polynomial in n. Note that using approximate local operations in the local search Procedure B (in Figure 2.1) makes the running time of the algorithm polynomial. The reason is as follows: One can easily show that for any ground set X of elements, the value of the initial set $S = \{v\}$ is at least $\mathsf{Opt}(X)/n$, where $\mathsf{Opt}(X)$ is the optimal value of problem (2.1) restricted to X. Each local operation in Procedure B increases the value of the function by a factor $1 + \frac{\epsilon}{n^4}$. Therefore, the number of local operations for Procedure B is at most $\log_{1+\frac{\epsilon}{n^4}} \frac{\mathsf{Opt}(X)}{\frac{\mathsf{Opt}(X)}{n}} = O(\frac{1}{\epsilon} n^4 \log n)$, and thus the running time of the whole procedure is $\frac{1}{\epsilon} \cdot n^{O(k)}$. Moreover, the number of procedure calls of Algorithm A for Procedure B is $k+1$, and thus the running time of Algorithm A is also polynomial.

Next, we prove the performance guarantee of Algorithm A. Let C denote the optimal solution to the original problem $\max\{f(S) \; : \; S \in \cap_{j=1}^k \mathcal{I}_j, \; S \subseteq V\}$. Let $C_i = C \cap V_i$ for each $i \in [k+1]$; so $C_1 = C$. Observe that C_i is a feasible solution to the problem (2.5) solved in the ith iteration. Applying Lemma 2.5 to problem (2.5) using the local optimum S_i and solution C_i, we obtain the following:

$$(1+\epsilon)(k+1) \cdot f(S_i) \geq f(S_i \cup C_i) + k \cdot f(S_i \cap C_i) \qquad \forall 1 \leq i \leq k+1. \tag{2.6}$$

Using $f(S) \geq \max_{i=1}^{k+1} f(S_i)$, we add these $k+1$ inequalities and simplify inductively as follows.

CLAIM 2.7. *For any $1 \leq l \leq k+1$, we have the following:*

$$(1+\epsilon)(k+1)^2 \cdot f(S) \geq (l-1) \cdot f(C) + f\left(\cup_{p=1}^{l} S_p \cup C_1\right) + \sum_{i=l+1}^{k+1} f(S_i \cup C_i) + \sum_{p=1}^{l-1}(k-l+p) \cdot f(S_p \cap C_p) + k \cdot \sum_{i=l}^{k+1} f(S_i \cap C_i).$$

Proof. We argue by induction on l. The base case $l = 1$ is trivial by just considering the sum of the $k+1$ inequalities in statement (2.6) above. Assuming the statement for some value $1 \leq l < k+1$, we prove the corresponding statement for $l+1$.

$$\begin{aligned}
(1+\epsilon)(k+1)^2 \cdot f(S) &\geq (l-1) \cdot f(C) + f\left(\cup_{p=1}^{l} S_p \cup C_1\right) \\
&\quad + \sum_{i=l+1}^{k+1} f(S_i \cup C_i) + \sum_{p=1}^{l-1}(k-l+p) f(S_p \cap C_p) + k \cdot \sum_{i=l}^{k+1} f(S_i \cap C_i) \\
&= (l-1) \cdot f(C) + f\left(\cup_{p=1}^{l} S_p \cup C_1\right) + f(S_{l+1} \cup C_{l+1}) \\
&\quad + \sum_{i=l+2}^{k+1} f(S_i \cup C_i) + \sum_{p=1}^{l-1}(k-l+p) f(S_p \cap C_p) + k \cdot \sum_{i=l}^{k+1} f(S_i \cap C_i) \\
&\geq (l-1) \cdot f(C) + f\left(\cup_{p=1}^{l+1} S_p \cup C_1\right) + f(C_{l+1}) \\
&\quad + \sum_{i=l+2}^{k+1} f(S_i \cup C_i) + \sum_{p=1}^{l-1}(k-l+p) f(S_p \cap C_p) + k \cdot \sum_{i=l}^{k+1} f(S_i \cap C_i) \\
&= (l-1) \cdot f(C) + f\left(\cup_{p=1}^{l+1} S_p \cup C_1\right) + f(C_{l+1}) + \sum_{p=1}^{l} f(S_p \cap C_p) \\
&\quad + \sum_{i=l+2}^{k+1} f(S_i \cup C_i) + \sum_{p=1}^{l}(k-l-1+p) f(S_p \cap C_p) + k \cdot \sum_{i=l+1}^{k+1} f(S_i \cap C_i) \\
&\geq l \cdot f(C) + f\left(\cup_{p=1}^{l+1} S_p \cup C_1\right) \\
&\quad + \sum_{i=l+2}^{k+1} f(S_i \cup C_i) + \sum_{p=1}^{l}(k-l-1+p) f(S_p \cap C_p) + k \cdot \sum_{i=l+1}^{k+1} f(S_i \cap C_i).
\end{aligned}$$

The first inequality is the induction hypothesis, and the next two inequalities follow from submodularity, using $(\cup_{p=1}^{l} S_p \cup C_1) \cap (S_{l+1} \cup C_{l+1}) = C_{l+1}$ and $(\cup_{p=1}^{l} S_p \cap C_p) \cup C_{l+1} = C$. ∎

Using the statement of Claim 2.7 when $l = k+1$, we obtain $(1+\epsilon)(k+1)^2 \cdot f(S) \geq k \cdot f(C)$. ∎

Finally, we give an improved approximation algorithm for symmetric submodular functions f that satisfy $f(S) = f(\overline{S})$ for all $S \subset V$. Symmetric submodular functions have been considered widely in the literature [17, 44], and it appears that symmetry allows for better approximation results and thus deserves separate attention.

THEOREM 2.8. *There is a $(\frac{1}{(1+\epsilon)(k+2)})$-approximation algorithm for maximizing nonnegative symmetric submodular functions subject to k matroid constraints.*

Proof. The algorithm for symmetric submodular functions is much simpler. In this case, we need only to perform *one* iteration of the approximate local search Procedure B (as opposed to $k+1$ in Theorem 2.6). Let C denote the optimal solution and S_1 the result of the local search (on V). Then Lemma 2.2 implies the following:

$$(1+\epsilon)(k+1)\cdot f(S_1) \geq f(S_1 \cup C) + k \cdot f(S_1 \cap C) \geq f(S_1 \cup C) + f(S_1 \cap C).$$

Because f is symmetric, we also have $f(S_1) = f(\overline{S_1})$. Adding these two,

$$(1+\epsilon)(k+2)\cdot f(S_1) \geq f(\overline{S_1}) + f(S_1 \cup C) + f(S_1 \cap C) \geq f(C \setminus S_1) + f(S_1 \cap C) \geq f(C)\ .$$

Thus we have the desired approximation guarantee. □

3. Knapsack constraints. In this section, we give an approximation algorithm for submodular maximization subject to multiple knapsack constraints. Let $f : 2^V \to \mathbb{R}_+$ be a submodular function and $w^1, \ldots, w^k$ be k weight-vectors corresponding to knapsacks having capacities $C_1, \ldots, C_k$ respectively. The problem we consider in this section is as follows:

$$\max \left\{ f(S)\ :\ \sum_{j \in S} w^i_j \leq C_i\ \forall 1 \leq i \leq k,\ S \subseteq V \right\}. \tag{3.1}$$

By scaling each knapsack, we assume that $C_i = 1$ for all $i \in [k]$; we also assume that all weights are rational. We denote $f_{max} = \max\{f(v)\ :\ v \in V\}$. We assume without loss of generality that for every $i \in V$, the singleton solution $\{i\}$ is feasible for all the knapsacks (otherwise such elements can be dropped from consideration). To solve the above problem, we first define a fractional relaxation of the submodular function and give an approximation algorithm for this fractional relaxation (see section 3.2). Then we show how to design an approximation algorithm for the original integral problem using the solution for the fractional relaxation (see section 3.3). Let $F : [0,1]^n \to \mathbb{R}_+$, the *fractional relaxation of* f, be the "extension-by-expectation" [53],

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

Note that F is a multilinear polynomial in variables $x_1, \ldots, x_n$ and has continuous derivatives of all orders. Furthermore, as shown in Vondrák [53], for all $i, j \in V$, $\frac{\partial^2}{\partial x_j \partial x_i} F \leq 0$ everywhere on $[0,1]^n$; we refer to this condition as *continuous submodularity.*

3.1. Extending function f on scaled ground sets. Let $s_i \in \mathbb{Z}_+$ be arbitrary values for each $i \in V$. Define a new ground set U that contains s_i "copies" of each element $i \in V$; so the total number of elements in U is $\sum_{i \in V} s_i$. We will denote any subset T of U as $T = \cup_{i \in V} T_i$, where each T_i consists of all copies of element $i \in V$ from T. Now define function $g : 2^U \to \mathbb{R}_+$ as $g(\cup_{i \in V} T_i) = F(\ldots, \frac{|T_i|}{s_i}, \ldots)$. We make use of the following useful property of g, which is Lemma 2.3 from Mirrokni, Schapira, and Vondrák [39].

LEMMA 3.1 (see [39]). *Set function g is a submodular function on ground set U.*

3.2. Solving the fractional relaxation. We now present an algorithm for obtaining a near-optimal fractional feasible solution for maximizing a nonnegative submodular function over k knapsack constraints. Let $w^1, \ldots, w^k$ denote the weight-vectors in each of the k knapsacks; recall that all knapsacks have capacity one. For ease of exposition, it is useful to consider a more general problem where each variable has additional upper bounds $\{u_i \in [0,1]\}_{i=1}^n$, i.e.,

$$\max\{F(y) \ : \ w^s \cdot y \leq 1 \ \ \forall s \in [k], \ \ 0 \leq y_i \leq u_i \ \ \forall i \in V\}. \tag{3.2}$$

We first define a local search procedure for problem (3.2) and prove some properties of it (see Lemmas 3.3 and 3.5). Then we present the approximation algorithm (see Figure 3.2) for solving the fractional relaxation when all upper bounds are one (see Theorem 3.6).

Estimating function F. It is not clear how to evaluate the fractional relaxation F exactly using a value-oracle to the original submodular function f. However, as shown in [53, 7], function F can be efficiently estimated within a small additive error; this is described below for completeness. Let $L \gg n^{13}$ be some large value. For any input $x \in [0,1]^n$, define *estimate*

$$\omega(x) := \frac{1}{L} \cdot \sum_{j=1}^{L} f(A_j),$$

where $\{A_j\}_{j=1}^L$ are independent samples, and each $A_j \subseteq V$ includes each element $i \in V$ independently with probability x_i. Clearly $\omega(x)$ can be evaluated using a polynomial number of value-queries to function f. Also observe that $\omega(x)$ is a random variable with expected value $F(x)$.

Define $f_{max} := \max\{f(\emptyset), \max\{f(\{v\}) : v \in V\}\}$; by subadditivity, $\max_{S\subseteq V} f(S) \leq n \cdot f_{max}$. Hence $F(x) \leq n \cdot f_{max}$ for any $x \in [0,1]^n$. The next claim shows that the estimates ω are close to F.

CLAIM 3.2. *For any $x \in [0,1]^n$, $Pr[|\omega(x) - F(x)| > \frac{f_{max}}{n^5}] \leq \exp(-n)$.*

Proof. Recall that $\omega(x) = \frac{1}{L}\sum_{j=1}^L f(A_j)$ where random sets $\{A_j\}_{j=1}^L$ are as defined above. Define random variables $Y_j := f(A_j)/(n \cdot f_{max})$; note that $Y_j \in [0,1]$ since we have $0 \leq f(S) \leq n \cdot f_{max}$ for all $S \subseteq V$. Also define $Y := \frac{L\omega(x)}{nf_{max}} = \sum_{j=1}^L Y_j$, a sum of independent $[0,1]$ random variables. Let $\mu := E[Y] = \frac{L}{nf_{max}}F(x)$ and $\delta := \frac{L}{\mu n^6}$. Note that $\mu \leq L$ since $F(x) \leq n \cdot f_{max}$. By Chernoff bounds [40], we have the following:

$$\Pr\left[|Y - \mu| > \frac{L}{n^6}\right] = \Pr[|Y - \mu| > \delta\mu] \leq 2 \cdot \exp\left(\frac{-\mu\delta^2}{2+\delta}\right) \leq \exp(-n). \tag{3.3}$$

The last inequality is by the following two cases. If $\delta \geq 1$, then $\frac{\mu\delta^2}{2+\delta} \geq \frac{\mu\delta}{3} = \frac{L}{3n^6} \geq 2n$. Else $\delta < 1$, and in this case $\frac{\mu\delta^2}{2+\delta} \geq \frac{\mu\delta^2}{3} = \frac{L^2}{3\mu n^{12}} \geq \frac{L}{3n^{12}} \geq 2n$. Now

$$\Pr\left[\left|\omega(x) - F(x)\right| > \frac{f_{max}}{n^5}\right] = \Pr\left[\left|\frac{L\omega(x)}{nf_{max}} - \frac{LF(x)}{nf_{max}}\right| > \frac{L}{n^6}\right] = \Pr\left[\left|Y - \mu\right| > \frac{L}{n^6}\right].$$

Combined with (3.3) we obtain the claim. ∎

Local search for problem (3.2). Denote the region $\mathcal{U} := \{y \ : \ 0 \leq y_i \leq u_i \ \forall i \in V\}$. For the local search, we consider only values for each variable from a discrete set of values in $[0,1]$, namely, $\mathcal{G} = \{p \cdot \zeta \ : \ p \in \mathbb{N}, \ 0 \leq p \leq \frac{1}{\zeta}\}$, where $\zeta = \frac{1}{10n^4}$. Using standard scaling methods, we assume (at the loss of $1 + o(1)$ factor in the optimal value of (3.2)) that all upper bounds $\{u_i\}_{i\in V} \subseteq \mathcal{G}$. We also assume

Input: Knapsack weights $\{w^s\}_{s=1}^k$, variable upper bounds $\{u_i \in [0,1]\}_{i=1}^n$, discretization $\mathcal{G}$, parameter ϵ, and value oracle access to submodular function f.

1. Set $a \leftarrow \arg\max\{u_a \cdot f(\{a\}) \mid a \in V\}$.
2. If $u_a \cdot f(\{a\}) \le f(\emptyset)$, set $y(i) \leftarrow 0$ for all $i \in V$; else set
$$y(i) = \begin{cases} u_a, & i = a, \\ 0, & i \in V \setminus \{a\}. \end{cases}$$
3. While the following local operation applies, update y accordingly.
   - Let subset $M \subseteq [n]$ with $|M| \le 2k$ and values $\{v_e \in \mathcal{G} \mid e \in M\}$. Define
$$y'(i) := \begin{cases} v_i, & i \in M, \\ y(i), & i \in V \setminus M. \end{cases}$$
   If the following conditions hold, then set $y \leftarrow y'$: (1) $y' \in \mathcal{U}$, (2) y' satisfies all knapsack constraints, and (3) the *estimate* $\omega(y') > (1+\epsilon)\cdot\omega(y)$.
4. **Output** y as the local optimum.

FIG. 3.1. *The approximate local search procedure for problem* (3.2).

that each upper bound is positive (otherwise the ground set of (3.2) can be reduced), and hence $u_i \ge \zeta$ for all $i \in V$. Let $\epsilon > 0$ be a parameter to be fixed later; ϵ will be inverse polynomial in n. The local search procedure for problem (3.2) is given in Figure 3.1. In the analysis, we assume that n is larger than any fixed constant (otherwise problem (3.1) is exactly solvable by enumeration).

LEMMA 3.3. *With high probability, the local search procedure (Figure* 3.1*) terminates in polynomial time, and the resulting local optimum $\tilde{y}$ satisfies $F(z) \le (1+\epsilon)\cdot F(\tilde{y}) + \frac{f_{max}}{n^4}$ for every $z \in \mathcal{G}^n$ in the local neighborhood of $\tilde{y}$.*

Proof. Observe that each local neighborhood in the algorithm in Figure 3.1 has size $N = n^{O(k)}$. Since each estimate ω can be evaluated using polynomially many queries to function f, it follows that each iteration takes $n^{O(k)}$ time which is polynomial.

We first show that the algorithm has at most $T := 6\log_2 n/\epsilon$ iterations whp. An estimate $\omega(x)$ for some input x is called a *success* if $|\omega(x) - F(x)| \le f_{max}/(n^5)$. By Claim 3.2, the first $N \cdot T$ estimates queried by the algorithm in Figure 3.1 are all successes with probability at least $1 - NT \cdot \exp(-n) = 1 - o(1)$ since both N and T are polynomial in n. In the rest of this proof we assume that the first $N \cdot T$ estimates made by the algorithm are successes; this event occurs with probability $1 - o(1)$.

Observe that the initial solution y_o chosen in step 2 satisfies $F(y_o) \ge \max\{u_a \cdot f(\{a\}), f(\emptyset)\}$, where a is the index chosen in step 1. Since each upper bound $u_a \ge \zeta$ (see above assumption), we have $F(y_o) \ge \zeta \cdot f_{max} = f_{max}/(10n^4)$. Now since $\omega(y_o)$ is a success, we also have $\omega(y_o) \ge F(y_o) - f_{max}/n^5 \ge f_{max}/(11n^4)$ for n large enough.

As observed earlier, $\max_{S\subseteq V} f(S) \le n \cdot f_{max}$. Hence for any $x \in [0,1]^n$, the estimate $\omega(x)$ is always at most $n \cdot f_{max}$. Since the observed ω-value increases by a $1+\epsilon$ factor in each iteration, the number of iterations of this local search is bounded by $\frac{\log(11n^5)}{\log(1+\epsilon)} \le \frac{4+5\log n}{\epsilon} \le T$ for large enough n. Since each iteration takes polynomial time, the algorithm terminates in polynomial time whp.

Let $\tilde{y}$ denote the resulting local optimum. Consider any $z \in \mathcal{G}^n$ in the local neighborhood of $\tilde{y}$ (cf. step 3 in Figure 3.1). By the local optimality condition, we have $\omega(z) \le (1+\epsilon)\cdot\omega(\tilde{y})$. Since we assume that each of the first NT estimates made by the algorithm are successes, we also have the following:

$$F(z) \le \omega(z) + \frac{f_{max}}{n^5} \le (1+\epsilon)\cdot\omega(\tilde{y}) + \frac{f_{max}}{n^5} \le (1+\epsilon)\cdot F(\tilde{y}) + \frac{3f_{max}}{n^5}.$$

The first and last inequalities follow from the fact that $\omega(z)$ and $\omega(\tilde{y})$ are successes, and the second inequality is by local optimality. This implies the lemma since n is large enough. ∎

Let $\tilde{y} \in \mathcal{U} \cap \mathcal{G}^n$ denote a local optimal solution obtained upon running the local search in Figure 3.1; in the following we assume that this solution satisfies Lemma 3.3, which happens whp. We first prove the following simple claim based on the discretization $\mathcal{G}$.

CLAIM 3.4. *Suppose $\alpha, \beta \in [0,1]^n$ are such that each has at most k positive coordinates $y' := \tilde{y} - \alpha + \beta \in \mathcal{U}$, and y' satisfies all knapsacks. Then $F(y') \leq (1+\epsilon) \cdot F(\tilde{y}) + \frac{1}{4n^2} f_{max}$.*

Proof. Let $z \in \mathcal{U} \cap \mathcal{G}^n$ be such that $z \leq y'$ coordinatewise and $\sum_{i=1}^n (y'_i - z_i)$ is minimized. Note that z is a feasible local move from $\tilde{y}$: It lies in $\mathcal{G}^n$, satisfies all knapsacks and the upper bounds, and is obtainable from $\tilde{y}$ by changing $2k$ variables (the positive coordinates in α and β). Hence by local optimality in Lemma 3.3, $F(z) \leq (1+\epsilon) \cdot F(\tilde{y}) + \frac{f_{max}}{n^4}$.

By the choice of z, it follows that $|z_i - y'_i| \leq \zeta$ for all $i \in V$. Suppose B is an upper bound on all first partial derivatives of function F on $\mathcal{U}$, i.e., $|\frac{\partial F(x)}{\partial x_i}|_{\bar{x}}| \leq B$ for all $i \in V$ and $\bar{x} \in \mathcal{U}$. Then because F has continuous derivatives, we obtain

$$|F(z) - F(y')| \leq \sum_{i=1}^n B \cdot |z_i - y'_i| \leq nB\zeta \leq 2n^2 f_{max} \cdot \zeta \leq \frac{f_{max}}{5n^2}.$$

The last inequality uses $\zeta = \frac{1}{10n^4}$, and the second to last inequality uses $B \leq 2n \cdot f_{max}$ which we show next. Consider any $\bar{x} \in [0,1]^n$ and $i \in V$. We have

$$\begin{aligned} \left| \frac{\partial F(x)}{\partial x_i} |_{\bar{x}} \right| &= \left| \sum_{S \subseteq [n] \setminus \{i\}} (f(S \cup \{i\}) - f(S)) \cdot \Pi_{a \in S} \bar{x}_a \cdot \Pi_{b \in S^c \setminus i} (1 - \bar{x}_b) \right| \\ &\leq \max_{S \subseteq [n] \setminus \{i\}} [f(S \cup \{i\}) + f(S)] \;\leq\; 2n \cdot f_{max} \;. \end{aligned}$$

Thus we have $F(y') \leq F(z) + \frac{1}{5n^2} f_{max} \leq (1+\epsilon) \cdot F(\tilde{y}) + \frac{f_{max}}{5n^2} + \frac{f_{max}}{n^4} \leq (1+\epsilon) \cdot F(\tilde{y}) + \frac{f_{max}}{4n^2}$ for large n. ∎

For any $x, y \in \mathbb{R}^n$, we define $x \vee y$ (meet operator) and $x \wedge y$ (join operator) by $(x \vee y)_j := \max(x_j, y_j)$ and $(x \wedge y)_j := \min(x_j, y_j)$, respectively, for $j \in [n]$.

LEMMA 3.5. *For locally optimal $\tilde{y} \in \mathcal{U} \cap \mathcal{G}^n$ and any $\tilde{x} \in \mathcal{U}$ satisfying the knapsack constraints, we have $(2 + 2n \cdot \epsilon) \cdot F(\tilde{y}) \geq F(\tilde{y} \wedge \tilde{x}) + F(\tilde{y} \vee \tilde{x}) - \frac{1}{2n} \cdot f_{max}$.*

Proof. For the sake of analysis, we add the following k *dummy elements* to the ground set: For each knapsack $s \in [k]$, element d_s has weight 1 in knapsack s and zero in all other knapsacks, and upper bound of 1. The function f remains the same: It depends only on the original variables V. Let $W := V \cup \{d_s\}_{s=1}^k$ denote the new ground set. Using the dummy elements, any fractional feasible solution can be augmented to another of the same F-value while satisfying all knapsacks at *equality*. We augment $\tilde{y}$ and $\tilde{x}$ using dummy elements to obtain y and x that both satisfy all knapsacks at equality. Clearly $F(y) = F(\tilde{y})$, $F(y \wedge x) = F(\tilde{y} \wedge \tilde{x})$, and $F(y \vee x) = F(\tilde{y} \vee \tilde{x})$. Let $y' = y - (y \wedge x)$ and $x' = x - (y \wedge x)$. Note that for all $s \in [k]$, $w^s \cdot y' = w^s \cdot x'$, and let $c_s = w^s \cdot x'$. We will decompose y' and x' into an equal number of terms as $y' = \sum_t \alpha_t$ and $x' = \sum_t \beta_t$ such that αs and βs have small support, and $w^s \cdot \alpha_t = w^s \cdot \beta_t$ for all t and $s \in [k]$.

1. Initialize $t \leftarrow 1$, $\gamma \leftarrow 1$, $x'' \leftarrow x'$, and $y'' \leftarrow y'$.

2. While $\gamma > 0$, do the following:
   (a) Consider $LP_x := \{z \geq 0 \ : \ z \cdot w^s = c_s \ \forall s \in [k]\}$ where the variables are restricted to indices $i \in [n]$ with $x''_i > 0$. Similarly $LP_y := \{z \geq 0 \ : \ z \cdot w^s = c_s \ \forall s \in [k]\}$ where the variables are restricted to indices $i \in [n]$ with $y''_i > 0$. Let $u \in LP_x$ and $v \in LP_y$ be extreme points.
   (b) Set $\delta_1 = \max\{\chi \ : \ \chi \cdot u \leq x''\}$, $\delta_2 = \max\{\chi \ : \ \chi \cdot v \leq y''\}$, and $\delta = \min\{\delta_1, \delta_2\}$.
   (c) Set $\beta_t \leftarrow \delta \cdot u$, $\alpha_t \leftarrow \delta \cdot v$, $\gamma \leftarrow \gamma - \delta$, $x'' \leftarrow x'' - \beta_t$, and $y'' \leftarrow y'' - \alpha_t$.
   (d) Set $t \leftarrow t + 1$.

We first show that this procedure is well-defined. A simple induction shows that at the start of every iteration, $w^s \cdot x'' = w^s \cdot y'' = \gamma \cdot c_s$ for all $s \in [k]$. Thus in step 2(a), LP_x (resp., LP_y) is nonempty: x''/γ (resp., y''/γ) is a feasible solution. From the definition of LP_x and LP_y it also follows that $\delta > 0$ in step 2(b) and at least one coordinate of x'' or y'' is zeroed out in step 2(c). This implies that the decomposition procedure terminates in $r \leq 2n$ steps.

At the end of the procedure, we have decompositions $x' = \sum_{t=1}^r \beta_t$ and $y' = \sum_{t=1}^r \alpha_t$. Furthermore, each α_t (resp., β_t) corresponds to an *extreme point* of LP_y (resp., LP_x) in some iteration: Hence the number of positive components in any of $\{\alpha_t, \beta_t\}_{t=1}^r$ is at most k, and all these values are rational. Finally, note that for all $t \in [r]$, $w^s \cdot \alpha_t = w^s \cdot \beta_t$ for all knapsacks $s \in [k]$. Note that x, y, x', y', αs, and βs are vectors over W.

For each $t \in [r]$, define $\tilde{\alpha}_t$ (resp., $\tilde{\beta}_t$) to be α_t (resp., β_t) restricted to the original variables V. From the above decomposition, it is clear that $\tilde{y} = \tilde{y} \wedge \tilde{x} + \sum_{t=1}^r \tilde{\alpha}_t$ and $\tilde{x} = \tilde{y} \wedge \tilde{x} + \sum_{t=1}^r \tilde{\beta}_t$, where $\tilde{\alpha}$s and $\tilde{\beta}$s are nonnegative. Thus for any $t \in [r]$, $\tilde{y} - \tilde{\alpha}_t + \tilde{\beta}_t \in \mathcal{U}$. Furthermore, for any $t \in [r]$, $y - \alpha_t + \beta_t \geq 0$ coordinatewise and satisfies all knapsacks at equality; hence dropping the dummy variables, we obtain that $\tilde{y} - \tilde{\alpha}_t + \tilde{\beta}_t$ satisfies all knapsacks (perhaps not at equality). Now observe that Claim 3.4 applies to $\tilde{y}$, $\tilde{\alpha}_t$, and $\tilde{\beta}_t$ (for any $t \in [r]$) because each of $\tilde{\alpha}_t, \tilde{\beta}_t$ has support-size at most k, and (as argued above) $\tilde{y} - \tilde{\alpha}_t + \tilde{\beta}_t \in \mathcal{U}$ and satisfies all knapsacks. Thus

$$F(\tilde{y} - \tilde{\alpha}_t + \tilde{\beta}_t) \leq (1 + \epsilon) \cdot F(\tilde{y}) + \frac{f_{max}}{4n^2} \qquad \forall t \in [r] \ . \tag{3.4}$$

Let $M \in \mathbb{Z}_+$ be large enough so that $M\tilde{\alpha}_t$ and $M\tilde{\beta}_t$ are integral for all $t \in [r]$. In the rest of the proof, we consider a scaled ground set U containing M copies of each element in V. We define function $g : 2^U \to \mathbb{R}_+$ as $g(\cup_{i \in V} T_i) = F(\ldots, \frac{|T_i|}{M}, \ldots)$, where each T_i consists of copies of element $i \in V$. Lemma 3.1 implies that g is submodular. Corresponding to $\tilde{y}$ we have a set $P = \cup_{i \in V} P_i$ consisting of the first $|P_i| = M \cdot \tilde{y}_i$ copies of each element $i \in V$. Similarly $\tilde{x}$ corresponds to set $Q = \cup_{i \in V} Q_i$ consisting of the first $|Q_i| = M \cdot \tilde{x}_i$ copies of each element $i \in V$. Hence $P \cap Q$ (resp., $P \cup Q$) corresponds to $\tilde{x} \wedge \tilde{y}$ (resp., $\tilde{x} \vee \tilde{y}$) scaled by M. Again, $P \setminus Q$ (resp., $Q \setminus P$) corresponds to a scaled version of $\tilde{y} - (\tilde{y} \wedge \tilde{x})$ (resp., $\tilde{x} - (\tilde{y} \wedge \tilde{x})$). The decomposition $\tilde{y} = (\tilde{y} \wedge \tilde{x}) + \sum_{t=1}^r \tilde{\alpha}_t$ from above suggests *disjoint* sets $\{A_t\}_{t=1}^r$ such that $\cup_t A_t = P \setminus Q$; i.e., each A_t corresponds to $\tilde{\alpha}_t$ scaled by M. Similarly there are *disjoint* sets $\{B_t\}_{t=1}^r$ such that $\cup_t B_t = Q \setminus P$. Observe also that $g((P \setminus A_t) \cup B_t) = F(\tilde{y} - \tilde{\alpha}_t + \tilde{\beta}_t)$, so (3.4) corresponds to

$$g((P \setminus A_t) \cup B_t) \leq (1 + \epsilon) \cdot g(P) + \frac{f_{max}}{4n^2} \qquad \forall t \in [r] \ .$$

Adding all these r inequalities, we obtain

$$r(1+\epsilon)\cdot g(P)+\frac{r}{4n^2}f_{max}\geq\sum_{t=1}^{r}g((P\setminus A_t)\cup B_t). \tag{3.5}$$

In the following calculation, for any $j\in\{0,1,\ldots,r\}$, let $A_{\leq j}:=\cup_{t=1}^{j}A_t$. Now

$$\begin{aligned}\sum_{t=1}^{r}g\left((P\setminus A_t)\bigcup B_t\right)&=\sum_{t=1}^{r}\left[g\left((P\setminus A_t)\bigcup B_t\right)+g\left(P\setminus A_{\leq t-1}\right)\right]-\sum_{t=1}^{r}g\left(P\setminus A_{\leq t-1}\right)\\&\geq\sum_{t=1}^{r}\left[g(P\cup B_t)+g\left(P\setminus A_{\leq t}\right)\right]-\sum_{t=1}^{r}g\left(P\setminus A_{\leq t-1}\right)\\&=\sum_{t=1}^{r}g(P\cup B_t)+g(P\setminus A_{\leq r})-g(P)\\&=\sum_{t=1}^{r}g(P\cup B_t)+g(P\cap Q)-g(P). \qquad (3.6)\end{aligned}$$

The inequality above is obtained by applying submodularity on all terms of the first summation, using the fact that family $\{A_t\}_{t=1}^{r}$ is disjoint, and the last equality uses $A_{\leq r}=P\setminus Q$. The summation in (3.6) can be simplified in a similar manner (below for any $j\in\{0,1,\ldots,r\}$, let $B_{\leq j}:=\cup_{t=1}^{j}B_t$).

$$\begin{aligned}\sum_{t=1}^{r}g(P\cup B_t)&=\sum_{t=1}^{r}\left[g\left(P\cup B_t\right)+g\left(P\cup B_{\leq t-1}\right)\right]-\sum_{t=1}^{r}g\left(P\cup B_{\leq t-1}\right)\\&\geq\sum_{t=1}^{r}\left[g(P\cup B_{\leq t})+g(P)\right]-\sum_{t=1}^{r}g\left(P\cup B_{\leq t-1}\right)\\&=(r-1)\cdot g(P)+g\left(P\cup B_{\leq r}\right). \qquad (3.7)\end{aligned}$$

This uses the fact that the B_ts's are disjoint. Recall that $B_{\leq t}=Q\backslash P$, which combined with (3.7), (3.6), and (3.5) implies

$$r(1+\epsilon)\cdot g(P)+\frac{r}{4n^2}f_{max}\geq(r-2)\cdot g(P)+g(P\cup Q)+g(P\cap Q).$$

Hence $(2+\epsilon r)\cdot g(P)\geq g(P\cup Q)+g(P\cap Q)-\frac{r}{4n^2}f_{max}$. This implies the lemma because $r\leq 2n$. ∎

Approximation algorithm for problem (3.2) with upper bounds one. The algorithm is given in Figure 3.2 and is similar to the way Algorithm A in section 2 uses the local search Procedure B.

THEOREM 3.6. *For any $\frac{1}{n}\ll\delta<\frac{1}{4}$, the algorithm in Figure* 3.2 *is a randomized* $(\frac{1}{4}-\delta)$*-approximation algorithm for the fractional knapsack problem* (3.2) *with all upper bounds $u_i=1$ (for all $i\in V$).*

Proof. The algorithm of Figure 3.2 runs in polynomial time whp, since it involves only two calls to the local search procedure (Figure 3.1) that runs in polynomial time (by Lemma 3.3 as $\epsilon\geq\frac{1}{n^3}$).

Since each singleton solution is feasible for the knapsacks and upper bounds are one, T_0 (in step 1) is a feasible solution of value f_{max}. Let x denote the globally optimal solution to the given instance of problem (3.2) (recall all upper bounds

Input: Knapsack weights $\{w^s\}_{s=1}^k$, parameter δ, and value oracle to submodular function f.

1. Set T_0 to be one of $\emptyset, \{1\}, \{2\}, \ldots, \{n\}$ having maximum f-value.
2. Choose $\epsilon \leftarrow \delta/8n$ as the parameter for local search (Figure 3.1).
3. Run the local search (Figure 3.1) with all upper bounds at 1, to get local optimum y_1.
4. Run the local search in Figure 3.1 again, with upper bound $1-y_1(i)$ for each $i \in [n]$, to obtain local optimum y_2.
5. **Output** $\arg\max\{f(T_0), F(y_1), F(y_2)\}$.

FIG. 3.2. *Approximation algorithm for the fractional knapsack problem.*

are 1). We will show $(2+\delta/4)\cdot(F(y_1)+F(y_2)) \geq F(x) - f_{max}/n$, which would prove the theorem, since this implies the following:

$$\begin{aligned}\max\left\{f_{max}, F(y_1), F(y_2)\right\} &\geq \frac{1}{2+\delta/8}F(y_1) + \frac{1}{2+\delta/8}F(y_2) + \frac{\delta/8}{2+\delta/8}f_{max}\\ &\geq \frac{F(x) - f_{max}/n}{(2+\delta/8)\cdot(2+\delta/4)} + \frac{\delta/8}{2+\delta/8}f_{max}\\ &\geq \frac{F(x)}{(2+\delta/8)\cdot(2+\delta/4)}\\ &\geq \frac{1}{4+\delta}F(x) \;\geq\; \left(\frac{1}{4}-\delta\right)F(x).\end{aligned}$$

Observe that x is a feasible solution in the first local search (step 3), and $x' = x - (x \wedge y_1)$ is a feasible solution to the second local search (step 4). Since we use $\epsilon = \delta/8n$, Lemma 3.5 implies that whp the following hold for the two local optima:

$$\left(2+\frac{\delta}{4}\right)\cdot F(y_1) \geq F(x\wedge y_1) + F(x \vee y_1) - \frac{f_{max}}{2n}\ ,$$
$$\left(2+\frac{\delta}{4}\right)\cdot F(y_2) \geq F(x'\wedge y_2) + F(x' \vee y_2) - \frac{f_{max}}{2n}\ .$$

We show that $F(x\wedge y_1)+F(x\vee y_1)+F(x'\vee y_2) \geq F(x)$, which suffices to prove the theorem. For this inequality, we again consider a scaled ground set U having M copies of each element in V (where $M \in \mathbb{Z}_+$ is large enough so that Mx, My_1, and My_2 are all integral). Define function $g : 2^U \to \mathbb{R}_+$ as $g(\cup_{i\in V} T_i) = F(\ldots, \frac{|T_i|}{M}, \ldots)$, where each T_i consists of copies of element $i \in V$. Lemma 3.1 implies that g is submodular. Also define the following subsets of U: A (representing y_1) consists of the first $My_1(i)$ copies of each element $i \in V$, C (representing x) consists of the first $Mx(i)$ copies of each element $i \in V$, and B (representing y_2) consists of $My_2(i)$ copies of each element $i \in V$ (namely, the copies numbered $My_1(i)+1$ through $My_1(i)+My_2(i)$) so that $A \cap B = \emptyset$. Note that we can indeed pick such sets because $y_1+y_2 \leq 1$ coordinatewise. Also we have the following correspondences via scaling:

$$A\cap C \equiv x \wedge y_1, \quad A \cup C \equiv x \vee y_1, \quad (C\setminus A)\cup B \equiv x' \vee y_2\ .$$

Thus it suffices to show $g(A\cap C) + g(A\cup C) + g((C\setminus A)\cup B) \geq g(C)$. But this follows from submodularity and nonnegativity of g:

$$g(A\cap C)+g(A\cup C)+g((C\setminus A)\cup B) \geq g(A\cap C)+g(C\setminus A)+g(C\cup A\cup B) \geq g(C).$$

Hence we have the desired approximation for the fractional problem (3.2). $\square$

3.3. Rounding the fractional knapsack. Figure 3.3 describes our algorithm for submodular maximization subject to k knapsack constraints (problem (3.1)).

Input: Knapsack weights $\{w^s\}_{s=1}^k$, parameter η, and value oracle to submodular function f.

1. Set $c \leftarrow \frac{16}{\eta}$, $\delta \leftarrow \frac{1}{8c^3k^4}$, and $\epsilon \leftarrow \frac{1}{ck}$.
2. Define an element $e \in V$ as *heavy* if $w^s(e) \geq \delta$ for some knapsack $s \in [k]$. All other elements are called *light*.
3. Enumerate over all feasible (under the knapsacks) sets consisting of up to k/δ heavy elements, to obtain T_1 having maximum f-value.
4. Restricting to only light elements, solve the fractional relaxation (problem (3.2)) with all upper bounds one, using the algorithm in Figure 3.2 (with parameter $\eta/3$). Let x denote the fractional solution found.
5. Obtain random set R as follows: Pick each light element $e \in V$ into R independently with probability $(1-\epsilon)x_e$.
6. If R satisfies all knapsacks, set $T_2 \leftarrow R$; otherwise set $T_2 \leftarrow \emptyset$.
7. **Output** $\arg\max\{f(T_1), f(T_2)\}$.

FIG. 3.3. *Approximation algorithm for submodular maximization under k knapsacks.*

The rest of this section proves the following theorem.

THEOREM 3.7. *For any constant $\eta > 0$, the algorithm in Figure* 3.3 *is a randomized $(\frac{1}{5} - \eta)$-approximation algorithm for maximizing nonnegative submodular functions over k knapsack constraints.*

Note that the running time of the algorithm in Figure 3.3 is polynomial for any fixed k: The enumeration in step 3 takes $n^{O(k/\delta)}$ time, and the algorithm in Figure 3.2 is also polynomial time.

We now analyze the performance guarantee. Let H and L denote the heavy and light elements, respectively, in an optimal solution. Note that $|H| \leq k/\delta$ since all knapsacks have capacity one. Hence enumerating over all possible sets of heavy elements in step 3, we obtain profit $f(T_1) \geq f(H)$.

We now focus only on light elements and show that the expected profit $f(T_2)$ is at least $\frac{1}{4} \cdot f(L)$. Let $\mathsf{Opt} \geq f(L)$ denote the optimal value of the problem considered in step 4. Theorem 3.6 implies that the resulting fractional solution satisfies $F(x) \geq (\frac{1}{4} - \frac{\eta}{3}) \cdot \mathsf{Opt}$, whp. Note that the definition of T_2 implies it is always feasible. In the following we lower bound the expected profit. Let $\alpha(R) := \max\{w^i(R) \ : \ i \in [k]\}$.

CLAIM 3.8. *For any $a \geq 1$, $\Pr[\alpha(R) \geq a] \leq k \cdot e^{-cak^2}$.*

Proof. Fixing a knapsack $i \in [k]$, we will bound $\Pr[w^i(R) \geq a]$. Let X_e denote the binary random variable which is set to 1 iff $e \in R$, and let $Y_e = \frac{w^i(e)}{\delta} X_e$. Because we deal only with light elements, each Y_e is a $[0,1]$ random variable. Let $Z_i := \sum_e Y_e$ and $\mu_i := E[Z_i] \leq \frac{1-\epsilon}{\delta}$. By scaling, it suffices to upper bound $\Pr[w^i(R) \geq a] = \Pr[Z_i \geq a/\delta]$. Because the Y_e's are independent $[0,1]$ random variables, Chernoff bounds [40] imply

$$\Pr[Z_i \geq a/\delta] \leq e^{-a\epsilon^2/8\delta} = e^{-cak^2}.$$

Finally, by a union bound, we obtain $\Pr[\alpha(R) \geq a] \leq \sum_{i=1}^k \Pr[w^i(R) \geq a] \leq k \cdot e^{-cak^2}$. □

CLAIM 3.9. *For any $a \geq 1$, $\max\{f(S) \ : \ S \subseteq$ light elements, $\alpha(S) \leq a+1\} \leq 2(1+\delta)k(a+1) \cdot \mathsf{Opt}$.*

Proof. We will show that for any set S with $\alpha(S) \leq a+1$, $f(S) \leq 2(1+\delta)k(a+1) \cdot \mathsf{Opt}$, which implies the claim. Consider partitioning set S into a number of smaller parts each of which satisfies all knapsacks as follows. As long as there are remaining elements in S, form a group by greedily adding S elements until no more addition is possible, and then continue to form the next group. Except for the last group formed, every other group must have filled up some knapsack to extent $1-\delta$ (otherwise another light element can be added). Thus the number of groups partitioning S is at most $\frac{k(a+1)}{1-\delta}+1 \leq 2k(a+1)(1+\delta)$. Because each of these groups is a feasible solution, the claim follows by the subadditivity of f. ∎

CLAIM 3.10. *In step 5, the expected value* $E[f(R)] \geq (1-\epsilon)\cdot F(x)$.

Proof. Recall that $E[f(R)] = F((1-\epsilon)x)$. Now function F is concave along any nonnegative direction vector [53]. In particular, the function $h(t) := F(t\cdot x)$, where $t \in [0,1]$ is concave (since $x \in [0,1]^n$). Thus

$$F((1-\epsilon)x) = h(1-\epsilon) \geq (1-\epsilon)\cdot h(1) + \epsilon\cdot h(0) \geq (1-\epsilon)\cdot h(1) = (1-\epsilon)\cdot F(x).$$

This implies the claim. ∎

LEMMA 3.11. *In step 6, the expectation* $E[f(T_2)] \geq (\frac{1}{4}-\eta)\cdot \mathsf{Opt}$.

Proof. Define the following disjoint events: $A_0 := \{\alpha(R) \leq 1\}$, and $A_\ell := \{\ell < \alpha(R) \leq 1+\ell\}$ for any $\ell \in \mathbb{N}$. Note that the expected value of $f(T_2)$ is $\mathsf{ALG} := E[f(S) \mid A_0]\cdot \Pr[A_0]$. We can write

$$\begin{aligned} E[f(R)] &= E[f(R) \mid A_0]\cdot \Pr[A_0] + \sum_{\ell\geq 1} E[f(R) \mid A_\ell]\cdot \Pr[A_\ell] \\ &= \mathsf{ALG} + \sum_{\ell \geq 1} E[f(R) \mid A_\ell]\cdot \Pr[A_\ell]. \end{aligned}$$

For any $\ell \geq 1$, from Claim 3.8 we have $\Pr[A_\ell] \leq \Pr[\alpha(R) > \ell] \leq k\cdot e^{-c\ell k^2}$. From Claim 3.9 we have $E[f(R) \mid A_\ell] \leq \max\{f(S) : S \subseteq \text{light elements}, \alpha(S) \leq \ell+1\} \leq 2(1+\delta)k(\ell+1)\cdot \mathsf{Opt}$. So

$$E[f(R) \mid A_\ell]\cdot \Pr[A_\ell] \leq k\cdot e^{-c\ell k^2}\cdot 2(1+\delta)k(\ell+1)\cdot \mathsf{Opt} \leq 8\cdot \mathsf{Opt}\cdot \ell k^2\cdot e^{-c\ell k^2}.$$

Consider the expression $\sum_{\ell\geq 1} \ell k^2\cdot e^{-c\ell k^2} \leq \sum_{t\geq 1} t\cdot e^{-ct} \leq \frac{1}{c}$ for large enough constant c. Thus

$$\begin{aligned} \mathsf{ALG} &= E[f(R)] - \sum_{\ell\geq 1} E[f(R) \mid A_\ell]\cdot \Pr[A_\ell] \\ &\geq E[f(R)] - 8\cdot \mathsf{Opt} \sum_{\ell \geq 1} \ell k\cdot e^{-c\ell k} \geq E[f(R)] - \frac{8}{c}\mathsf{Opt}. \end{aligned}$$

By Claim 3.10 and Theorem 3.6 (and using $\epsilon \leq \eta/2$),

$$E[f(R)] \geq (1-\epsilon)\cdot F(x) \geq (1-\epsilon)\cdot\left(\frac{1}{4}-\frac{\eta}{3}\right)\cdot\mathsf{Opt} \geq \left(\frac{1}{4}-\frac{\eta}{3}-\frac{\epsilon}{4}\right)\cdot \mathsf{Opt} \geq \left(\frac{1}{4}-\frac{\eta}{2}\right)\cdot\mathsf{Opt}.$$

Combining the above two equations (using $\eta = \frac{16}{c}$), we obtain $\mathsf{ALG} \geq \left(\frac{1}{4}-\eta\right)\cdot \mathsf{Opt}$. ∎

Completing the proof of Theorem 3.7. Recall that H and L denote the heavy and light elements, respectively, in an optimal integral solution; so the optimal value is $f(H \cup L)$. The enumeration procedure (step 3) for heavy elements produces a solution T_1 with $f(T_1) \geq f(H)$. Lemma 3.11 implies that the rounding procedure for light elements (step 6) produces solution T_2 with $E[f(T_2)] \geq (\frac{1}{4}-\eta)\cdot f(L)$. The expected value obtained by the algorithm is $E[\max\{f(T_1), f(T_2)\}] \geq \max\{f(T_1), E[f(T_2)]\}$ which is at least

$$\max\left\{f(H), \left(\frac{1}{4}-\eta\right)\cdot f(L)\right\} \geq \frac{1}{5}\cdot f(H) + \frac{4}{5}\cdot\left(\frac{1}{4}-\eta\right)\cdot f(L) \geq \left(\frac{1}{5}-\eta\right)\cdot f(H\cup L).$$

The last inequality uses $f(H \cup L) \leq f(H) + f(L)$ by subadditivity. This implies the desired approximation guarantee in Theorem 3.7.

4. Improved bounds under partition matroids. In this section, we consider a special case of problem (2.1) when all the underlying matroids are partition matroids. In this case we obtain improved approximation ratios for both monotone and nonmonotone submodular functions.

The algorithm for partition matroids is again based on local search. In the *exchange* local move of the general case (section 2), the algorithm attempts only to include one new element at a time (while dropping up to k elements). Here we generalize that step to allow including p new elements while dropping up to $k \cdot p$ elements for some fixed constant $p \geq 1$. Specifically, given a current solution $S \in \cap_{j=1}^{k}\mathcal{I}_j$, the local moves to consider are as follows:

- *Delete operation.* If $e \in S$ such that $f(S \setminus \{e\}) > f(S)$, then $S \leftarrow S \setminus \{e\}$.
- *p-exchange operation.* For some $q \leq p$, if $d_1, \ldots, d_q \in V \setminus S$ and $e_i \in S \cup \{\emptyset\}$ (for $1 \leq i \leq k\cdot q$) are such that (i) $S' = (S\setminus\{e_i\ :\ 1 \leq i \leq kq\}) \cup \{d_1, \ldots, d_q\} \in \mathcal{I}_j$ for all $j \in [k]$ and (ii) $f(S') > f(S)$, then $S \leftarrow S'$.

The main idea here is the following strengthening of Lemma 2.2.

LEMMA 4.1. *For a local optimal solution S under deletions and p-exchanges, and any $C \in \cap_{j=1}^{k}\mathcal{I}_j$, we have $k \cdot f(S) \geq (1-\frac{1}{p}) \cdot f(S \cup C) + (k-1) \cdot f(S \cap C)$.*

Proof. We use an exchange property (see Schrijver [48]), which implies for any *partition matroid* $\mathcal{M}$ and $C, S \in \mathcal{I}(\mathcal{M})$ the existence of a map $\pi : C\setminus S \to (S\setminus C) \cup \{\emptyset\}$ such that

1. $(S \setminus \{\pi(b)\ :\ b \in T\}) \cup T \in \mathcal{I}(\mathcal{M})$ for all $T \subseteq C \setminus S$;
2. $|\pi^{-1}(e)| \leq 1$ for all $e \in S \setminus C$.

Let π_j denote the mapping under partition matroid $\mathcal{M}_j$ (for $1 \leq j \leq k$). For any subset $T \subseteq C \setminus S$ and $j \in [k]$, we denote $\pi_j(T) := \{\pi_j(e) \mid e \in T\}$.

Combining partition matroids $\mathcal{M}_1$ and $\mathcal{M}_2$. We use π_1 and π_2 to construct a multigraph G on vertex set $C\setminus S$ and edge-set labeled by $E = \pi_1(C\setminus S) \cup \pi_2(C\setminus S) \subseteq S \setminus C$ as follows:

- For each $a \in \pi_1(C \setminus S) \setminus \pi_2(C \setminus S)$, graph G has a loop (u, u) labeled a, where $u \in C \setminus S$ is the unique element with $\pi_1(u) = a$.
- For each $b \in \pi_2(C \setminus S) \setminus \pi_1(C \setminus S)$, graph G has a loop (u, u) labeled b, where $u \in C \setminus S$ is the unique element with $\pi_2(u) = b$.
- For each $e \in \pi_1(C \setminus S) \cap \pi_2(C \setminus S)$, graph G has an edge (u, v) labeled e, where $u, v \in C \setminus S$ are the unique elements with $\pi_1(u) = \pi_2(v) = e$.

Note that each edge in G has a unique label, and the maximum number of edges incident to any vertex is two. Hence G is a vertex-disjoint union of cycles and paths, where loops appear only at end-points of paths. We index vertices of G (i.e., elements of $C\setminus S$) using $\{1, 2, \ldots, |C\setminus S|\}$ in such a way that vertices along any path or cycle in G are numbered consecutively. For any $q \in \{0, \ldots, p-1\}$, let R_q denote the elements of $C \setminus S$ having an index that is *not* q modulo p. It is clear that the induced graph $G[R_q]$ for any $q \in [p]$ consists of disjoint paths/cycles (possibly with loops at path end-points), where each path/cycle has length at most p. Furthermore, each element of $C \setminus S$ appears in exactly $p-1$ sets among $\{R_q\}_{q=0}^{p-1}$.

CLAIM 4.2. *For any* $q \in \{0, \ldots, p-1\}$, $k \cdot f(S) \geq f(S \cup R_q) + (k-1) \cdot f(S \cap C)$.

Proof. The following arguments hold for any $q \in [p]$, and for notational simplicity we denote $R = R_q \subseteq C \setminus S$. Let $\{D_l\}_{l=1}^t$ denote the vertices in connected components of $G[R]$ that form a partition of R. As mentioned above, $|D_l| \leq p$ for all $l \in [t]$. For any $l \in [t]$, let E_l denote the labels of edges in G *incident* to vertices D_l (i.e., edges with at least one end-point in D_l). Because $\{D_l\}_{l=1}^t$ are distinct connected components in $G[R]$, sets $\{E_l\}_{l=1}^t$ are disjoint subsets of $E \subseteq S \setminus C$. Note also that by G's structure, $|E_l| \leq |D_l| + 1$.

Consider any $l \in [t]$; we claim that $S_l = (S \setminus E_l) \cup D_l \in \mathcal{I}_1 \cap \mathcal{I}_2$. By the construction of graph G, we have $E_l \supseteq \{\pi_1(b) \ : \ b \in D_l\}$ and $E_l \supseteq \{\pi_2(b) \ : \ b \in D_l\}$. Hence $S_l \subseteq (S \setminus \{\pi_i(b) \ : \ b \in D_l\}) \cup D_l$ for $i = 1, 2$. But from the property of mapping π_i (where $i = 1, 2$), $(S \setminus \{\pi_i(D_l)\}) \cup D_l \in \mathcal{I}_i$. This implies that $S_l \in \mathcal{I}_1 \cap \mathcal{I}_2$ for all $l \in [t]$ as claimed.

From the properties of the maps π_j for each partition matroid $\mathcal{M}_j$, we have $(S \setminus \pi_j(D_l)) \cup D_l \in \mathcal{I}_j$ for each $3 \leq j \leq k$. Thus the following sets are independent in all matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$:

$$\left(S \setminus (\cup_{j=3}^k \pi_j(D_l) \cup E_l)\right) \cup D_l \qquad \forall\ l \in [t].$$

Define $A_l := \left(\cup_{j=3}^k \pi_j(D_l)\right) \cup E_l \subseteq S \setminus C$ for each $l \in [t]$; note that $|A_l| \leq (k-1) \cdot p + 1$. Recall that $\{E_l\}_{l=1}^t$ are disjoint subsets of $S \setminus C$. So using the property of mappings π_js, each element $i \in S \setminus C$ appears in $n_i \leq k-1$ of the sets $\{A_l\}_{l=1}^t$. Since $|D_l| \leq p$ and $|A_l| \leq (k-1) \cdot p + 1$ (for any $l \in [t]$), the local optimality of S implies

$$f(S) \geq f\left(\left(S \setminus A_l\right) \cup D_l\right) \qquad \forall l \in [t].$$

Adding these inequalities and simplifying using submodularity and disjointness of $\{D_l \subseteq C \setminus S\}_{l=1}^t$,

$$(t+1) \cdot f(S) \geq \sum_{l=1}^t f(S \setminus A_l) + f\left(S \cup (\cup_{l=1}^t D_l)\right). \tag{4.1}$$

Using local optimality under deletions, we have the inequalities

$$(k-1-n_i) \cdot f(S) \geq (k-1-n_i) \cdot f(S \setminus \{i\}) \qquad \forall i \in S \setminus C. \tag{4.2}$$

Combining inequalities (4.1) and (4.2),

$$\begin{aligned} f\left(S \cup (\cup_{l=1}^t D_l)\right) - f(S) &\leq \sum_{l=1}^t [f(S) - f(S \setminus A_l)] \\ &\quad + \sum_{i \in S \setminus C} (k-1-n_i) \cdot [f(S) - f(S \setminus \{i\})] \\ &= \sum_{l=1}^{\lambda} [f(S) - f(S \setminus T_l)], \end{aligned} \tag{4.3}$$

where $\lambda := t + \sum_{i \in S \setminus C} (k-1-n_i)$ and $\{T_l\}_{l=1}^{\lambda}$ are subsets of $S \setminus C$ such that each element of $S \setminus C$ appears in exactly $k-1$ of them. Thus we can simplify expression (4.3) using Claim 2.3 to obtain

$$f\left(S \cup \left(\cup_{l=1}^t D_l\right)\right) - f(S) \leq (k-1) \cdot (f(S) - f(S \cap C)).$$

Noting that $\cup_{l=1}^{t} D_l = R$, we have the claim. $\square$

Adding the p inequalities given by Claim 4.2, we get $pk \cdot f(S) \geq \sum_{q=0}^{p-1} f(S \cup R_q) + p(k-1) \cdot f(S \cap C)$. Note that $\{S \cup R_q\}_{q=0}^{p-1}$ are subsets of $S \cup C$ such that each element of $C \setminus S$ is missing in exactly one set and elements of S are missing in none of them. Rearranging this inequality and applying Claim 2.3,

$$p(k-1) \cdot f(S \cap C) + p \cdot f(S \cup C) - pk \cdot f(S) \leq \sum_{q=0}^{p-1} [f(S \cup C) - f(S \cup R_q)] \leq f(S \cup C) - f(S).$$

Thus

$$(pk-1) \cdot f(S) \geq (p-1) \cdot f(S \cup C) + p(k-1) \cdot f(S \cap C),$$

which implies $k \cdot f(S) \geq (1 - \frac{1}{p}) \cdot f(S \cup C) + (k-1) \cdot f(S \cap C)$, giving the lemma. $\square$

Again, to ensure polynomial runtime of the local search, we define an approximate local search procedure identical to the one in Figure 2.1, except that we use the deletion and p-exchange local moves. Each iteration in this local search increases the f-value by a factor of $1 + \epsilon$. Similar to Lemma 2.5, it is easy to use Lemma 4.1 to prove the following.

LEMMA 4.3. *For an approximately locally optimal solution S in Procedure B (Figure 2.1) under deletions and p-exchange, and any $C \in \cap_{j=1}^{k} \mathcal{I}_j$, $(1+\epsilon)k \cdot f(S) \geq (1 - \frac{1}{p}) \cdot f(S \cup C) + (k-1) \cdot f(S \cap C)$.*

We are now ready to give the improved approximation guarantee under partition matroids.

THEOREM 4.4. *For any $k \geq 2$ and fixed constant $\delta > 0$, there exists a $\frac{1}{k+1+\frac{1}{k-1}+\delta}$-approximation algorithm for maximizing a nonnegative submodular function over k partition matroids. This bound improves to $\frac{1}{k+\delta}$ for monotone submodular functions.*

Proof. We set $p = 1 + \lceil \frac{8k}{\delta} \rceil$ and $\epsilon = \frac{\delta}{4k}$ for the approximate local search. The algorithm for the monotone case is just the local search procedure with p-exchanges. Lemma 4.3 applied to local optimal S and the global optimal C implies $(1 + \delta/4) \cdot f(S) \geq (\frac{1}{k} - \frac{1}{pk}) \cdot f(S \cup C) \geq (\frac{1}{k} - \frac{1}{pk}) \cdot f(C)$ (by nonnegativity and monotonicity). From the setting of p, solution S is a $k + \delta$ approximate solution.

For the nonmonotone case, the algorithm is identical to Algorithm A (Figure 2.2); again the approximate local search uses deletions and p-exchanges. If C denotes a global optimum, an identical analysis as in Theorem 2.6 yields

$$(1+\epsilon)\left(1 + \frac{1}{p-1}\right) k^2 \cdot f(S) \geq (k-1) \cdot f(C).$$

This uses the following inequalities implied by Lemma 4.3:

$$(1+\epsilon)\left(\frac{p}{p-1}\right) k \cdot f(S_i) \geq f(S_i \cup C_i) + (k-1) \cdot f(S_i \cap C_i) \qquad \forall 1 \leq i \leq k,$$

where S_i denotes the local optimal solution in iteration $i \in \{1, \ldots, k\}$ and $C_i = C \setminus \cup_{j=1}^{i-1} S_j$. Using the values of p and ϵ, solution S is a $(k + 1 + \frac{1}{k-1} + \delta)$-approximate solution.

Finally, observe that the algorithm has running time which is polynomial for fixed k and δ. $\square$

Tight example for greedy algorithm for monotone functions. We note that the result for monotone submodular functions is the first improvement over the greedy $\frac{1}{k+1}$-approximation algorithm [42], even for the special case of partition matroids. It is easy to see that the greedy algorithm is a $\frac{1}{k}$-approximation for *modular* functions. But it is only a $\frac{1}{k+1}$-approximation for monotone submodular functions. The following example shows that this bound is tight for every $k \geq 1$. Consider a ground set $E = \{e \ : \ 0 \leq e \leq p(k+1)+1\}$ of natural numbers (for $p \geq 2$ arbitrarily large); we define a family $\mathcal{F} = \{S_0, S_1, \ldots, S_k, T_1, T_2\}$ of $k+3$ subsets of E. We have $S_0 = \{e \ : \ 0 \leq e \leq p\}$, $T_1 = \{e \ : \ 0 \leq e \leq p-1\}$, $T_2 = \{p\}$, and for each $1 \leq i \leq k$, $S_i = \{e \ : \ p{\cdot}i{+}1 \leq e \leq p{\cdot}(i{+}1)\}$. The submodular function f is the coverage function defined on a family $\mathcal{F}$ of sets. That is, for any subset $X \subseteq \mathcal{F}$, $f(X)$ equals the number of elements in E covered by X; f is clearly monotone submodular. We now define k partition matroids over $\mathcal{F}$: for $1 \leq j \leq k$, the jth partition matroid has $\{S_0, S_j\}$ in one group (with bound one) and all other sets in singleton groups (each with bound one). In other words, the partition constraints require that for every $1 \leq j \leq k$, at most one of S_0 and S_j be chosen. Observe that $\{S_i \ : \ 1 \leq i \leq k\} \cup \{T_1, T_2\}$ is a feasible solution of value $|E| = p(k+1)+1$. However, the greedy algorithm picks S_0 first (because it has maximum size) and gets only value $p+1$.

5. Matroid base constraints.

A base in a matroid is any maximal independent set. In this section, we consider the problem of maximizing a nonnegative submodular function over *bases* of some matroid $\mathcal{M}$:

$$\max\{f(S) \ : \ S \in \mathcal{B}(\mathcal{M})\}. \tag{5.1}$$

We first consider the case of symmetric submodular functions.

THEOREM 5.1. *There is a $(\frac{1}{3} - \epsilon)$-approximation algorithm for maximizing a nonnegative symmetric submodular function over bases of any matroid.*

Proof. We use the natural local search algorithm based only on swap operations. The algorithm starts with any maximal independent set and performs improving *swaps* until none is possible. From the second statement of Lemma 2.2, if S is a local optimum and C is the optimal base, we have $2 \cdot f(S) \geq f(S \cup C) + f(S \cap C)$. Adding to this inequality the fact that $f(S) = f(\overline{S})$ using symmetry, we obtain $3 \cdot f(S) \geq f(S \cup C) + f(\overline{S}) + f(S \cap C) \geq f(C \setminus S) + f(S \cap C) \geq f(C)$. Using an approximate local search procedure to make the running time polynomial, we obtain the theorem. ☐

However, the approximation guarantee of this algorithm can be arbitrarily bad if the function f is not symmetric. An example is the directed-cut function in a *simple* digraph with a vertex bipartition (U, V) with $|U| = |V| = n$, having $t \gg 1$ edges from each U-vertex to V and 1 edge from each V-vertex to U. The matroid in this example is just the uniform matroid with rank n. It is clear that the optimal base is U; on the other hand, V is a local optimum under swaps.

We are not aware of a constant approximation for the problem of maximizing a submodular function subject to an arbitrary matroid base constraint. For a special class of matroids we obtain the following.

THEOREM 5.2. *There is a $(\frac{1}{6} - \epsilon)$-approximation algorithm for maximizing any nonnegative submodular function over bases of matroid $\mathcal{M}$ when $\mathcal{M}$ contains at least two disjoint bases.*

Proof. Let C denote the optimal base. The algorithm here first runs the local search algorithm using only swaps to obtain a base S_1 that satisfies $2 \cdot f(S_1) \geq f(S_1 \cup C) + f(S_1 \cap C)$ from Lemma 2.2. Then the algorithm runs a local search on

$V \setminus S_1$ using both exchanges and deletions to obtain an independent set $S_2 \subseteq V \setminus S_1$ satisfying $2 \cdot f(S_2) \geq f(S_2 \cup (C \setminus S_1)) + f(S_2 \cap (C \setminus S_1))$.

Consider the matroid $\mathcal{M}'$ obtained by contracting S_2 in $\mathcal{M}$. Note that by the assumption, there are two disjoint bases, say, A_1 and A_2, in the original matroid $\mathcal{M}$. Then $B_1 := A_1 \setminus \mathsf{cl}(S_2)$ and $B_2 := A_2 \setminus \mathsf{cl}(S_2)$ are disjoint bases in $\mathcal{M}'$; here cl denotes the closure operation [43]. Furthermore, B_1 and B_2 can also be computed in polynomial time.

Now observe that $S_2 \cup B_1$ and $S_2 \cup B_2$ are bases in the original matroid $\mathcal{M}$. The algorithm outputs solution S which is the better of the three bases: S_1, $S_2 \cup B_1$, and $S_2 \cup B_2$. We have

$$\begin{aligned} 6f(S) &\geq 2f(S_1) + 2\left(f(S_2 \cup B_1) + f(S_2 \cup B_2)\right) \geq 2f(S_1) + 2f(S_2) \\ &\geq f(S_1 \cup C) + f(S_1 \cap C) + f(S_2 \cup (C \setminus S_1)) \geq f(C). \end{aligned}$$

The second inequality uses the disjointness of B_1 and B_2. Finally, the approximate local search procedure can be used to ensure polynomial runtime, which implies the theorem. $\square$

A consequence of this result is the following.

COROLLARY 5.3. *Given any nonnegative submodular function $f : 2^V \to \mathbb{R}_+$ and an integer $0 \leq c \leq |V|$, there is a $(\frac{1}{6} - \epsilon)$-approximation algorithm for the problem $\max\{f(S) \ : \ S \subseteq V, \ |S| = c\}$.*

Proof. If $c \leq |V|/2$, then the assumption in Theorem 5.2 holds for the rank c uniform matroid, and the theorem follows. We show that $c \leq |V|/2$ can be ensured without loss of generality. Define function $g : 2^V \to \mathbb{R}_+$ as $g(T) = f(V \setminus T)$ for all $T \subseteq V$. Because f is nonnegative and submodular, so is g. Furthermore, $\max\{f(S) \ : \ S \subseteq V, \ |S| = c\} = \max\{g(T) \ : \ T \subseteq V, \ |T| = |V| - c\}$. Clearly one of c or $|V| - c$ is at most $|V|/2$, and we can apply Theorem 5.2 to the corresponding problem. $\square$

Acknowledgment. The proof of Lemma 2.2 presented in this paper is due to Jan Vondrák. Our original proof [35] was more complicated—we thank Jan for letting us present this simplified proof.

REFERENCES

[1] A. Ageev, R. Hassin, and M. Sviridenko, *A* 0.5*-approximation algorithm for MAX DICUT with given sizes of parts*, SIAM J. Discrete Math., 14 (2001), pp. 246–255.

[2] A. Ageev and M. Sviridenko, *An* 0.828 *approximation algorithm for the uncapacitated facility location problem*, Discrete Appl. Math., 93 (1999), pp. 149–156.

[3] A. Ageev and M. Sviridenko, *Pipage rounding: A new method of constructing algorithms with proven performance guarantee*, J. Comb. Optim., 8 (2004), pp. 307–328.

[4] P. Alimonti, *Non-oblivious local search for MAX* 2*-CCSP with application to MAX DICUT*, in Proceedings of the 23rd International Workshop on Graph-theoretic Concepts in Computer Science, 1997.

[5] K. M. Anstreicher, M. Fampa, J. Lee, and J. Williams, *Using continuous nonlinear relaxations to solve constrained maximum-entropy sampling problems*, Math. Program. Ser. A., 85 (1999), pp. 221–240.

[6] S. Burer and J. Lee, *Solving maximum-entropy sampling problems using factored masks*, Math. Program., 109 (2007), pp. 263–281.

[7] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrak, *Maximizing a submodular set function subject to a matroid constraint*, SIAM J. Comput., to appear.

[8] V. Cherenin, *Solving some combinatorial problems of optimal planning by the method of successive calculations*, in Proceedings of the Conference of Experiences and Perspectives of the Applications of Mathematical Methods and Electronic Computers in Planning, Mimeograph, Novosibirsk, 1962, (in Russian).

[9] G. Cornuéjols, M. Fischer, and G. Nemhauser, *Location of bank accounts to optimize oat: An analytic study of exact and approximation algorithms*, Management Sci., 23 (1977), pp. 789–810.

[10] G. Cornuéjols, M. Fischer, and G. Nemhauser, *On the uncapacitated location problem*, Ann. Discrete Math., 1 (1977), pp. 163–178.

[11] G. P. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey, *The uncapacitated facility location problem*, in Discrete Location Theory, Wiley, New York, 1990, pp. 119–171.

[12] J. Edmonds, *Matroids, submodular functions, and certain polyhedra*, in Combinatorial Structures and Their Applications, Gordon and Breach, New York, 1970, pp. 69–87.

[13] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[14] U. Feige and M. X. Goemans, *Approximating the value of two-prover systems, with applications to MAX-2SAT and MAX-DICUT*, in Proceedings of the 3rd Israel Symposium on Theory and Computing Systems, Tel Aviv, IEEE, 1995, pp. 182–189.

[15] U. Feige, V. Mirrokni, and J. Vondrák, *Maximizing nonmonotone submodular functions*, in Proceedings of the IEEE Foundations of Computer Science, 2007, pp. 461–471.

[16] A. Frank, *Matroids and submodular functions*, in Annotated Biblographies in Combinatorial Optimization, Wiley, New York, 1997, pp. 65–80.

[17] S. Fujishige, *Canonical decompositions of symmetric submodular systems*, Discrete Appl. Math., 5 (1983), pp. 175–190.

[18] M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni, *Approximating submodular functions everywhere*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2009, pp. 535–544.

[19] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[20] B. Goldengorin, G. Sierksma, G. Tijsssen, and M. Tso, *The data correcting algorithm for the minimization of supermodular functions*, Management Sci., 45 (1999), pp. 1539–1551.

[21] B. Goldengorin, G. Tijsssen, and M. Tso, *The Maximization of Submodular Functions: Old and New Proofs for the Correctness of the Dichotomy Algorithm*, SOM Report, University of Groningen, Groningen, The Netherlands, 1999.

[22] E. Halperin and U. Zwick, *Combinatorial approximation algorithms for the maximum directed cut problem*, in Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, 2001, pp. 1–7.

[23] J. Hartline, V. Mirrokni, and M. Sundararajan, *Optimal marketing strategies over social networks*, in Proceedings of the World Wide Web Conference (WWW), Microsoft, 2008, pp. 189–198.

[24] E. Hazan, S. Safra, and O. Schwartz, *On the complexity of approximating* k*-set packing*, Comput. Complexity, 15 (2006), pp. 20–39.

[25] J. Håstad, *Some optimal inapproximability results*, J. ACM, 48 (2001), pp. 798–869.

[26] S. Iwata, L. Fleischer, and S. Fujishige, *A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions*, J. ACM, 48 (2001), pp. 761–777.

[27] V. R. Khachaturov, *Mathematical Methods of Regional Programming*, Nauka, Moscow, 1989 (in Russian).

[28] A. Kulik, H. Shachnai, and T. Tamir, *Maximizing submodular functions subject to multiple linear constraints*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2009.

[29] C.-W. Ko, J. Lee, and M. Queyranne, *An exact algorithm for maximum entropy sampling*, Oper. Res., 43 (1995), pp. 684–691.

[30] A. Krause, A. Singh, and C. Guestrin, *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*, J. Mach. Learn. Res., 9 (2008), pp. 235–284.

[31] H. Lee, G. Nemhauser, and Y. Wang, *Maximizing a submodular function by integer programming: Polyhedral results for the quadratic case*, European J. Oper. Res., 94 (1996), pp. 154–166.

[32] J. Lee, *Maximum entropy sampling*, in Encyclopedia of Environmetrics, A. H. El-Shaarawi and W. W. Piegorsch, eds., Wiley, Chichester, 2001, pp. 1229–1233.

[33] J. Lee, *Semidefinite programming in experimental design*, in Handbook of Semidefinite Programming, Internat. Ser. Oper. Res. Management Sci. 27, H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., Springer, New York, 2000.

[34] J. Lee, *Constrained maximum-entropy sampling*, Oper. Res., 46 (1998), pp. 655–664.

[35] J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko, *Maximizing Nonmonotone Submodular Functions under Matroid and Knapsack Constraints*, IBM Research Report RC24679, 2008.

[36] J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko, *Nonmonotone submodular maximization under matroid and knapsack constraints*, in Proceedings of the ACM Symposium on Theory of Computing, 2009, pp. 323–332.

[37] J. Lee, M. Sviridenko, and J. Vondràk, *Submodular maximization over multiple matroids via generalized exchange properties*, in Proceedings of APPROX-RANDOM 2009, pp. 244–257.

[38] L. Lovász, *Submodular functions and convexity*, in Mathematical Programmming: The State of the Art, A. Bachem, M. N. Korte, and M. Grotschel, eds., Springer, New York, 1993, pp. 235–257.

[39] V. Mirrokni, M. Schapira, and J. Vondrák, *Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions*, in Proceedings of the ACM Conference on Electronic Commerce, 2008, pp. 70–77.

[40] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, 1995.

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

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

[43] J. G. Oxley, *Matroid Theory*, Oxford Sci. Publ., The Clarendon Press Oxford University Press, New York, 1992.

[44] M. Queyranne, *A combinatorial algorithm for minimizing symmetric submodular functions*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 1995, pp. 98–101.

[45] J. Reichel and M. Skutella, *Evolutionary algorithms and matroid optimization problems*, in Proceedings of the 9th ACM Genetic and Evolutionary Computation Conference (GECCO'07), 2007, pp. 947–954.

[46] T. G. Robertazzi and S. C. Schwartz, *An accelated sequential algorithm for producing D-optimal designs*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 341–358.

[47] A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, J. Combin. Theory Ser. B, 80 (2000), pp. 346–355.

[48] A. Schrijver, *Combinatorial Optimization,* Vols. A–C. Springer, Berlin, 2003.

[49] A. Schulz and N. Uhan, *Encouraging Cooperation in Sharing Supermodular Costs*, in Proceedings of APPROX-RANDOM, 2007, pp. 271–285

[50] M. C. Shewry and H. P. Wynn, *Maximum entropy sampling*, J. Appl. Stat., 14 (1987), pp. 165–170.

[51] M. Sviridenko, *A note on maximizing a submodular set function subject to knapsack constraint*, Oper. Res. Lett., 32 (2004), pp. 41–43.

[52] Z. Svitkina and L. Fleischer, *Submodular approximation: Sampling-based algorithms and lower bounds*, in Proceedings of IEEE Foundations of Computer Science, 2008, pp. 697–706.

[53] J. Vondrák, *Optimal approximation for the submodular welfare problem in the value oracle model*, in Proceedings of the ACM Symposium on Theory of Computing, 2008, pp. 67–74.

[54] J. Vondrák, *Symmetry and approximability of submodular maximization problems*, in Proceedings of IEEE Foundations of Computer Science, 2009.

[55] J. Vondrák, *personal communication*, 2008.