Polyhedral results for a class of cardinality constrained submodular minimization problems

Jiajin Yu and Shabbir Ahmed*
Georgia Institute of Technology, Atlanta, GA 30332

August 28, 2014

Abstract

Motivated by concave cost combinatorial optimization problems, we study the following mixed integer nonlinear set: $\mathcal{P} = \{(w, x) \in \mathbb{R} \times \{0,1\}^n : w \geq f(a'x), e'x \leq k\}$ where $f : \mathbb{R} \mapsto \mathbb{R}$ is a concave function, n and k are positive integers, $a \in \mathbb{R}^n$ is a nonnegative vector, $e \in \mathbb{R}^n$ is a vector of ones, and $x'y$ denotes the scalar product of vectors x and y of same dimension. A standard linearization approach for $\mathcal{P}$ is to exploit the fact that $f(a'x)$ is submodular with respect to the binary vector x. We extend this approach to take the cardinality constraint $e'x \leq k$ into account and provide a full description of the convex hull of $\mathcal{P}$ when the vector a has identical components. We also develop a family of facet-defining inequalities when the vector a has nonidentical components. Computational results using the proposed inequalities in a branch-and-cut framework to solve mean-risk knapsack problems show significant decrease in both time and the number of nodes over standard methods.

1 Introduction

Optimization problems in various applications involving economies of scale or risk averse behavior can be formulated as concave cost combinatorial optimization problems of the form

$$\text{(CCO)}: \ \min\left\{\sum_{\ell=1}^{L} f_\ell(a'_\ell x) : \ x \in X \subseteq \{0,1\}^n\right\},$$

where $f_\ell : \mathbb{R} \mapsto \mathbb{R}$ is concave and $a_\ell \in \mathbb{R}^n$ for $\ell = 1, \ldots, L$. The set X denotes combinatorial constraints, for example, those modeling feasible paths, assignments or knapsack solutions. Some specific examples of (CCO) are mentioned next.

Concave cost facility location: The concave cost facility location problem is an extension of the usual facility location problem to model economies of scale that can be achieved by connecting multiple customers to the same facility. With m customers and n facilities, the problem can be formulated as

$$\min\left\{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} f_j(\sum_{i=1}^{m} d_i x_{ij}) : \ \sum_{j=1}^{n} x_{ij} = 1 \ \forall\, i, \ x_{ij} \in \{0,1\} \ \forall\, i,j\right\},$$

where c_{ij} is the cost of assigning customer i to facility j, d_i is the demand of customer i, and f_j is a non-decreasing concave cost function modeling the economies of scale of serving demand from facility j. This (CCO) problem generalizes the well-known fixed-charge facility location problem and has been studied in [9,16,24,25].

*Corresponding Author. Email: sahmed@isye.gatech.edu

Mean-risk combinatorial optimization: Consider a stochastic combinatorial optimization problem $\min\{\tilde{c}'x : x \in X \subseteq \{0,1\}^n\}$ where the cost vector has independently distributed components $\tilde{c}_i$ with mean μ_i and variance ν_i for $i = 1, \ldots, n$. A typical deterministic equivalent formulation is to optimize a weighted combination of the mean and standard deviation of the cost, i.e.

$$\min\left\{\mu'x + \lambda\sqrt{\nu'x} :\ x \in X \subseteq \{0,1\}^n\right\},$$

where λ is a nonnegative weight. Such mean-risk combinatorial optimization problems in various settings have been studied in [3,18,21,23].

Approximate submodular minimization: Various combinatorial optimization problems with nondecreasing submodular cost functions of the form

$$\min\{F(x) :\ x \in X \subseteq \{0,1\}^n\}$$

have been considered (cf. [5,12,14,19]). Often the cost function F is only available through a value oracle. It has been shown [13] that a general nondecreasing submodular function F can be approximated upto a factor of $\sqrt{n}\log n$ by a function of the form $f(x) = \sqrt{c'x}$ using only a polynomial number of function value queries, i.e. $f(x) \le F(x) \le O(\sqrt{n}\log n)f(x)$ for all $x \in \{0,1\}^n$. Thus we can write an explicit formulation for an approximate version of the above submodular cost combinatorial optimization problem as the (CCO) problem:

$$\min\left\{\sqrt{c'x} :\ x \in X \subseteq \{0,1\}^n\right\}.$$

With a view towards developing mixed integer linear programming (MILP) based approaches for (CCO) we consider the following reformulation

$$\text{(CCO')} :\ \min\left\{\sum_{\ell=1}^{L} w_\ell :\ x \in X \subseteq \{0,1\}^n, (w_\ell, x) \in \operatorname{conv}(\mathcal{Q}_\ell)\forall\ \ell = 1, \ldots, L\right\},$$

where each set $\mathcal{Q}_\ell$ for $\ell = 1, \ldots, L$ is of the form

$$\mathcal{Q} = \{(w, x) \in \mathbb{R} \times \{0,1\}^n :\ w \ge f(a'x)\}, \tag{1}$$

and f is a concave function. Note that since $\mathcal{Q}$ is the union of a finite set of half lines with a common recession direction, its convex hull, denoted by $\operatorname{conv}(\mathcal{Q})$, is a polyhedron. When the vector a is nonnegative (or nonpositive) , the function $F(x) := f(a'x)$ is submodular over the binary hypercube. A function of binary variables $F : \{0,1\}^n \to \mathbb{R}$ is submodular if for all $x, y \in \{0,1\}^n$ with $x \le y$ (component-wise) and for some $i \in \{1, \ldots, n\}$ with $x_i = y_i = 0$, we have $F(x + e^i) - F(x) \ge F(y + e^i) - F(y)$, where $e^i \in \mathbb{R}^n$ is the i-th unit vector, i.e. the marginal values are diminishing [27]. By the classical results of Edmonds [8] and Lovasz [20] on submodular functions, we know an explicit inequality description of $\operatorname{conv}(\mathcal{Q})$. Thus (CCO') provides a mixed integer linear programming formulation of the mixed integer nonlinear program (CCO). In this paper we improve such a formulation by incorporating information from the constraints $x \in X$ in the set Q. In particular we study the following mixed-integer set

$$\mathcal{P} = \{(w, x) \in \mathbb{R} \times \{0,1\}^n : w \ge f(a'x), e'x \le k\}, \tag{2}$$

where f is a concave function, $a \in \mathbb{R}^n_+, k \in \mathbb{Z}_+$ and e is a vector of ones, and the cardinality constraint $e'x \le k$ is assumed to be implied by the constraints $x \in X$.

The contributions of this paper are as follows. First, we give a complete description of $\operatorname{conv}(\mathcal{P})$ when the vector a in (2) has identical components. Similar to the classical work in [8], we write a pair of primal-dual linear programs corresponding to the convex lower envelope of $f(a'x)$ and valid inequalities of $\operatorname{conv}(\mathcal{P})$ respectively. Then we construct a pair of optimal primal and dual solutions that provide the explicit inequality description of $\operatorname{conv}(\mathcal{P})$. Second, we give a family of facet-defining inequalities of $\operatorname{conv}(\mathcal{P})$ for general nonnegative a. We obtain such inequalities through sequence dependent exact lifting and present a polynomial time algorithm to find the lifting coefficients corresponding to any given sequence. We give another family of approximately lifted inequalities that can be weaker than the exactly lifted facet-defining inequalities but much faster to compute within a branch-and-cut procedure. Finally, we demonstrate the effectiveness of using the proposed inequalities in a branch-and-cut framework to solve mean-risk knapsack problems. Our computational results show significant decrease in both time and the number of nodes over standard methods.

We close this section by a brief discussion of related literature. As mentioned earlier, an inequality description of the convex hull of the epigraph of a general submodular function follows from the classical works [8, 20]. In the presence of constraints, submodular minimization is in general NP-hard and in most cases very hard to approximate [12, 19, 26]. Even for the cardinality constraint, the special case of size constrained minimum cut problem is NP-hard since it can be reduced to graph partitioning problem [10]. This suggest that there is little hope of obtaining a tractable inequality description of the convex hull of the epigraph of a submodular function under a cardinality constrained. However the submodular function considered here is of the special form $F(x) = f(a'x)$ where f is a concave function. Hassin and Tamir [17] solve the problem $\min\{c'x + f(a'x) : \ e'x \leq k, x \in \{0,1\}^n\}$ where f is a concave function, in polynomial time by reducing it to a two-dimension parametric linear programming problem. Onn and Rothblum [22] generalize the univariate concave function to multivariate case. From the equivalence of separation and optimization this suggests that for such a special submodular function a tractable description of the convex hull of the epigraph under a cardinality constraint is possible. However, to the best of our knowledge an explicit inequality description of such a set has not been presented before. Recently a number of works have used valid inequalities exploiting the submodularity of underlying functions within mixed integer linear programming approaches for various classes of mixed integer nonlinear programs [1–3, 5]. In particular, the work of Atamtürk and Narayanan [3] is most related to our work. They use the inequalities describing the convex hull of the epigraph of a submodular function whithout constraints, i.e. the set $\text{conv}(\mathcal{Q})$, to strengthen a second order cone programming (SOCP) relaxation of a mean-risk optimization problem in a branch-and-cut procedure and show that facet-defining inequalities of $\text{conv}(\mathcal{Q})$ significantly improves the performance. Our work extends this approach to include information from a cardinality constraint.

2 Valid Inequalities for $\text{conv}(\mathcal{P})$

In this section, we study valid inequalities of the convex hull of the set (2):

$$\mathcal{P} = \left\{(w,x) \in \mathbb{R} \times \{0,1\}^n \,:\, w \geq f(a'x),\, e'x \leq k\right\},$$

where f is a concave function and a is a nonnegative vector. The following notation will be used throughout. Let $N = \{1, \ldots, n\}$. Denote $a(S) := \sum_{i \in S} a_i$ for any $S \subseteq N$. Given a permutation $\sigma := ((1), \ldots, (n))$ of N, let $A_{(i)} = \sum_{j=1}^{i} a_{(j)}$ for all $(i) \in \sigma$.

Recall the unconstrained version of $\mathcal{P}$, i.e. the set $\mathcal{Q} = \{(w,x) \in \mathbb{R} \times \{0,1\}^n \,:\, w \geq f(a'x)\}$. Using the fact that $f(a'x)$ is submodular over $\{0,1\}^n$ it is known [8, 20] that every facet of $\text{conv}(\mathcal{Q})$, except the trivial bounds on x, is of the form

$$f(0) + \sum_{(i) \in \sigma} \rho_{(i)} x_{(i)} \leq w. \tag{3}$$

where

$$\rho_{(i)} = f(A_{(i-1)} + a_{(i)}) - f(A_{(i-1)}) \tag{4}$$

for $i \geq 1$ and σ is some permutation of N. Thus a complete inequality description of $\text{conv}(\mathcal{Q})$ is given by the inequalities (3) corresponding to all permutations. Although there is an exponential number of such inequalities, they can be separated in polynomial time. Following [3] we refer to the inequalities (3) as *extended polymatroid inequalities* (EPI). Clearly EP inequalities are valid for $\text{conv}(\mathcal{P})$. Next we strengthen the EP inequalities using the information from the cardinality constraint in $\mathcal{P}$.

Given a $(w,x) \in \mathbb{R} \times [0,1]^n$ with $x_{(1)} \geq x_{(2)} \geq \cdots \geq x_{(n)}$ we can check if $(w,x) \in \text{conv}(\mathcal{P})$ by solving the the following (dual) linear program:

$$\begin{aligned} \max\ & \pi_0 + \sum_{i=1}^{n} \pi_{(i)} x_{(i)} \\ \text{s.t.}\ & \pi_0 + \sum_{(i)\in S} \pi_{(i)} \le f(a(S))\ \ \forall S, |S| \le k. \end{aligned} \tag{5}$$

It follows that, given any permutation $\sigma = \{(1), \dots, (n)\}$, an inequality $\pi_0 + \sum \pi_{(i)} x_{(i)} \le w$ is valid for $\text{conv}(\mathcal{P})$ if and only if π_0, π is a feasible solution of (5). Also notice that coefficients of the EP inequality defined by (4) form a feasible solution of (5) since the EP inequality is valid for $\text{conv}(\mathcal{P})$.

Thus we can get an inequality description of $\text{conv}(\mathcal{P})$ if we obtained all optimal solutions to (5) corresponding to all vectors x. The construction is complicated by the fact that, unlike the unconstrained case, an optimal solution to (5) depends on the specific value of the vector x rather than just the permutation implied by its components. In the following subsection we show that we can do this construction and obtain a description of $\text{conv}(\mathcal{P})$ when all components of a are identical which we refer to as the *unweighted* case. In Section 2.2 we use a different approach, lifting, to derive valid inequalities for the *weighted* case where the vector a has nonidentical components.

2.1 Unweighted case: Identical Components

In this section we consider the case where $a_1 = \cdots = a_n = a$. Without loss of generality, unless otherwise specified, we consider the permutation $(1, \dots, n)$. We simplify the notation by denoting $f(A_i) = f(a \cdot i)$ as $f(i)$ and write the coefficients (4) of the EP inequality as $\rho_i = f(i) - f(i-1)$ for $i \in N$. We also denote $f(a(S)) = f(|S|)$ for any $S \subseteq N$.

Given a vector $x \in \{x \in [0,1]^n : e'x \le k\}$ such that $x_1 \ge x_2 \cdots \ge x_n$ we will next construct an optimal solution to (5). First we define a *critical* index $i_0 \in \{0, 1, \dots, k-1\}$ corresponding to x as follows. Let $x_0 := 1$ and $z_i := (k-i)x_i - \sum_{j=i+1}^{k-1} x_j$ for $i \in \{0, 1, \dots, k\}$ so that $z_k = 0, z_{k-1} = x_{k-1}, z_{k-2} = 2x_{k-2} - x_{k-1}, \dots, z_0 = k - \sum_{j=1}^{k-1} x_j$. Let $y := \sum_{j=k}^{n} x_j$ and

$$i_0 := \text{argmax}\,\{0 \le i \le k-1 \,:\, z_{i+1} \le y \le z_i\}\,. \tag{6}$$

Note that i_0 is well defined since the range $[z_{i+1}, z_i]$ is valid for any $i \in \{0, 1, \dots, k\}$ as $z_i - z_{i+1} = (k-i)(x_i - x_{i+1}) \ge 0$, and y will be in some interval $[z_{i+1}, z_i]$. The latter follows from the fact that $y \ge z_k = 0$ and $y \le z_0$ since $z_0 - y = k - \sum_{i=1}^{n} x_i \ge 0$ because x satisfies $e'x \le k$. Given $i_0 \in \{0, \dots, k-1\}$, we construct the following solution to (5):

$$\pi_j := \begin{cases} f(0) & j = 0 \\ \rho_j & j \le i_0 \\ \dfrac{f(k) - f(i_0)}{k - i_0} & j > i_0. \end{cases} \tag{7}$$

We will first show that the solution constructed above corresponding to any $i_0 \in \{0, \dots, k-1\}$ is feasible to (5) . Then we will show that if i_0 is constructed as in (6) then it is optimal. We will need the following result.

Lemma 2.1.1. *Given a concave function $f : \mathbb{R} \to \mathbb{R}$, for positive integers $t_1 > t_2, t_3 > t_4$, $t_2 \ge t_4$ and $t_1 - t_2 \ge t_3 - t_4$, we have*

$$\frac{f(t_1) - f(t_2)}{t_1 - t_2} \le \frac{f(t_3) - f(t_4)}{t_3 - t_4}.$$

Proof. For any integer i, we define $\rho_i = f(i) - f(i-1)$. Then we have

$$\begin{aligned}\frac{f(t_3)-f(t_4)}{t_3-t_4} &= \frac{\sum_{j\geq t_4}^{t_3-1}\rho_j}{t_3-t_4} \\ &\geq \frac{\sum_{j=t_2}^{t_2+t_3-t_4-1}\rho_j}{t_3-t_4} \\ &\geq \frac{\sum_{j=t_2}^{t_2+t_3-t_4-1}\rho_j + \sum_{j=t_2+t_3-t_4}^{t_1-1}\rho_j}{t_3-t_4+t_1-t_2-t_3+t_4} \\ &= \frac{f(t_1)-f(t_2)}{t_1-t_2},\end{aligned}$$

where the first inequality follows from the fact $\rho_j \geq \rho_{j'}$ when $j \leq j'$ (by concavity) and $t_2 \geq t_4$; and the second inequality follows from the same fact and $t_1 - t_2 \geq t_3 - t_4$. ☐

Proposition 2.1.2. *The solution π in* (7) *corresponding to any $i_0 \in \{0, \ldots, k-1\}$ is feasible for problem* (5).

Proof. For an $i_0 \in \{0, \ldots, k-1\}$, consider an arbitrary set S such that $|S| \leq k$. Let $S_1 = S \cap \{1, \ldots, i_0\}$ and $S_2 = S \backslash S_1$. Let $i_1 = |S_1|, i_2 = |S_2|$. Then from the construction (7), π satisfies

$$\pi_0 + \sum_{i\in S}\pi_i = \pi_0 + \sum_{i\in S_1}\pi_i + \sum_{i\in S_2}\pi_i \leq f(i_1) + \sum_{i\in S_2}\pi_i = f(i_1) + i_2\cdot\frac{f(k)-f(i_0)}{k-i_0}.$$

The first inequality in the above chain follows from the fact that for $i \in S_1$, π_is are coefficients of EPI and hence satisfies $\pi_0 + \sum_{i\in S_1}\pi_i \leq f(S_1) = f(i_1)$. The second equality follows from the definition of the dual solution in (7). If $i_2 = 0$, then we already know it is feasible. Assume $i_2 > 0$, we need to show

$$f(i_1) + i_2\cdot\frac{f(k)-f(i_0)}{k-i_0} \leq f(i_1+i_2),$$

which is equivalent to

$$\frac{f(k)-f(i_0)}{k-i_0} \leq \frac{f(i_1+i_2)-f(i_1)}{i_2}. \qquad (*)$$

We consider two cases:

$i_2 \leq k - i_0$**:** Inequality (*) follows from Lemma 2.1.1 by setting $t_1 = k$, $t_2 = i_0$, $t_3 = i_1 + i_2$ and $t_4 = i_1$. Note that $t_2 \geq t_4$ since $i_0 \geq i_1$ and $t_1 - t_2 \geq t_3 - t_4$ since $k - i_0 \geq i_2$.

$i_2 > k - i_0$**:** First by concavity, we know

$$i_2(f(k) - f(i_1+i_2)) \leq i_2(f(i) - f(i_1+i_2-(k-i_0))).$$

We also have

$$(i_2+i_0-k)(f(i_1+i_2)-f(i_1)) \leq i_2(f(i_1+i_2-(k-i_0))-f(i_1))$$

from Lemma 2.1.1 by setting $t_1 = i_1 + i_2, t_2 = i_1, t_3 = i_1 + i_2 - k + i_0$ and $t_4 = i_1$. To check that the condition of Lemma 2.1.1 is satisfied, first note $t_3 = i_1 + i_2 - (k - i_0) \geq 0$, then $t_2 = t_4$, and finally $t_1 - t_2 = i_2 \geq t_3 - t_4 = i_2 + i_0 - k$ since $i_0 \leq k$. Add these two inequalities together, and after rearrangement, we get inequality (*). ☐

Next we show that when i_0 is constructed as in (6) then π constructed as in (7) is an optimal solution to problem (5). We proceed by constructing a complementary solution to the following primal problem corresponding to (5):

$$
\begin{aligned}
\min \quad & \sum_{|S|\le k} P(S) f(|S|) \\
\text{s.t.} \quad & \sum_{S: i\in S, |S|\le k} P(S) = x_i \quad \forall i \in N \\
& \sum_{|S|\le k} P(S) = 1 \\
& P(S) \ge 0, \quad \forall S, |S| \le k.
\end{aligned} \tag{8}
$$

We construct a solution to (8) as follows:

$$
P(\emptyset) = 1 - x_1;\ P(\{1,\dots j\}) = x_j - x_{j+1}\ \forall j < i_0;\ \text{and}\ P(\{1,\dots,i_0\}) = \frac{z_{i_0} - y}{k - i_0}. \tag{9}
$$

Note that if $i_0 = 0$, then

$$
P(\emptyset) = \frac{z_0 - y}{k}.
$$

Let $\mathcal{S} := \{S \,:\, S \subset \{i_0+1,\dots,n\}, |S| = k - i_0\}$. For sets $S \in \mathcal{S}$ we set $P(S)$ according to those Lemma 2.1.3 below. All remaining sets $S \subseteq N$ with $|S| \le k$ are assigned $P(S) = 0$.

Lemma 2.1.3. *Suppose i_0 is constructed as in* (6) *then the following linear system in $P(S)$ for $S \in \mathcal{S}$*

$$
\sum_{S: i\in S} P(\{1,\dots,i_0\} \cup S) = x_i,\ i \in \{i_0+1,\dots,n\}
$$

has a nonnegative solution and

$$
\sum_{S\in\mathcal{S}} P(\{1,\dots,i_0\} \cup S) = \frac{\sum_{i=i_0+1}^{n} x_i}{k - i_0}.
$$

Proof. See Appendix. □

Lemma 2.1.4. *Suppose i_0 is constructed as in* (6) *then the solution $P(S)$ defined by* (9) *and Lemma 2.1.3 is a feasible solution to the primal problem* (8).

Proof. First we check that $\sum_{S: i\in S, |S|\le k} P(S) = x_i$ for all $i \in N$. If $i > i_0$, this holds by Lemma 2.1.3. For $i \le i_0$, we have (9), Lemma 2.1.3, and the definitions of z_{i_0} and y:

$$
\begin{aligned}
\sum_{j=i}^{i_0} P(\{1,\dots,j\}) + \sum_{S\in\mathcal{S}} P(S \cup \{1,\dots,i_0\}) &= x_i - x_{i_0} + \frac{z_{i_0} - y}{k - i_0} + \frac{\sum_{j=i_0+1}^{n} x_j}{k - i_0} \\
&= x_i - x_{i_0} + \frac{(k-i_0)x_{i_0} - \sum_{j=i_0+1}^{n} x_j}{k - i_0} + \frac{\sum_{j=i_0+1}^{n} x_j}{k - i_0} \\
&= x_i.
\end{aligned}
$$

Next we check that the $P(S)$'s sum up to one. For $i_0 > 0$, we have by the same steps as above for $i = 1$ and the construction of $P(\emptyset)$:

$$
P(\emptyset) + \sum_{j=1}^{i_0} P(\{1,\dots,j\}) + \sum_{S\in\mathcal{S}} P(S \cup \{1,\dots,i_0\}) = 1 - x_1 + x_1 = 1.
$$

For $i_0 = 0$, we have

$$
P(\emptyset) + \sum_{S\in\mathcal{S}} P(S) = \frac{k - \sum_{j=1}^{n} x_j}{k} + \frac{\sum_{j=1}^{n} x_j}{k} = 1.
$$

□

Proposition 2.1.5. *Suppose i_0 is constructed as in* (6) *then the solution π constructed as in* (7) *is an optimal solution for problem* (5).

Proof. By Lemma 2.1.2 and Lemma 2.1.4, we already know we have a pair of primal and dual feasible solutions. Now we verify that the objective value of the primal solution is the same as the dual solution. For $i_0 > 0$, we have

$$\begin{aligned} & f(\emptyset)P(\emptyset) + \sum_{l=1}^{i_0} P(\{1,\dots,l\})f(l) + \sum_{S\in\mathcal{S}} P(S\cup\{1,\dots,i_0\})f(k) \\ =& f(0) + \sum_{j=1}^{i_0}(f(j)-f(j-1))x_j + \sum_{j=i_0+1}^{n} \frac{f(k)-f(i_0)}{k-i_0}x_j. \end{aligned}$$

For $i_0 = 0$, we have

$$\begin{aligned} & f(\emptyset)P(\emptyset) + \sum_{S\in\mathcal{S}} P(S)f(k) \\ =& \frac{k-\sum_{j=1}^n x_j}{k} f(0) + \frac{\sum_{j=1}^n x_j}{k} f(k) \\ =& f(0) + \sum_{j=1}^{n} \frac{f(k)-f(0)}{k} x_j \end{aligned}$$

□

Theorem 2.1.6. *When $a_1 = \cdots = a_n = a$ then $\operatorname{conv}(\mathcal{P})$ is defined by the trivial inequalities $x \in [0,1]^n$, $e'x \le k$ and the following inequalities*

$$f(0) + \sum_{j=1}^{i_0}(f(j)-f(j-1))x_{(j)} + \sum_{j=i_0+1}^{n} \frac{f(k)-f(i_0)}{k-i_0}x_{(j)} \le w \tag{10}$$

corresponding to every permutation $\sigma = \{(1),\dots,(n)\}$ of N and $i_0 \in \{0,1,\dots,k-1\}$, where $f(j) = f(a\cdot j)$ for $j \in N$. Moreover given a $x \in [0,1]^n$, we can decide whether $x \in \operatorname{conv}(\mathcal{P})$ and find a violated inequality in $O(n\log n)$ time.

Proof. The first part follows from Propositions 2.1.2 and 2.1.5. Given $x \in [0,1]^n$ we can first check in $O(n)$ if $e'x \le k$. If yes, then to compute the coefficients of (10), first we sort the components of x, which takes $O(n\log n)$ time. Then finding the desired i_0 takes $O(n)$ time. Once i_0 is found, the coefficients can be computed according to (7) in $O(n)$ time. Therefore we can check for a violated inequality of the form (10) in $O(n\log n)$ time. □

We refer to the inequalities (10) as *separation inequalities* (SI) since they can be exactly separated.

2.2 Weighted case: Nonindentical Components

Now we consider the more general case where the components of the nonnegative vector a are not necessarily identical. We derive a family of facet-defining inequalities of $\operatorname{conv}(\mathcal{P})$ through lifting. Consider a set $S \subset N$ such that $|S| = k$. Without loss of generality, we consider the set $S = \{1,\dots,k\}$. The restriction of $\mathcal{P}$ by setting $x_i = 0$ for all $i \in N\backslash S$ is denoted by:

$$\mathcal{P}^0 = \left\{(w,x) \in \mathbb{R}\times\{0,1\}^S \,:\, w \ge f(a'x)\right\}.$$

Since there is no constraint on x in $\mathcal{P}^0$, we know that the extend polymatroid inequality (3) is facet-defining for $\mathcal{P}^0$:

$$f(0) + \sum_{i=1}^{k}\rho_i x_i \le w \tag{11}$$

We then lift variables $x_{k+1}, \ldots, x_n$ sequentially in that order. The intermediate set of feasible points are:

$$\mathcal{P}^i = \left\{(w, x) \in \mathbb{R} \times \{0,1\}^{\{1,\ldots,k,\ldots,i\}} \,:\, w \geq f(a'x),\ e'x \leq k\right\}\ \forall\, i = k+1, \ldots, n.$$

Given a facet-defining inequality

$$f(0) + \sum_{j=1}^{k} \rho_j x_j + \sum_{j>k}^{i-1} \zeta_j x_j \leq w$$

for $\text{conv}(\mathcal{P}^{i-1})$, the lifting problem associated with $\mathcal{P}^i$ is:

$$\begin{aligned} \zeta_i := \min \quad & w - f(0) - \sum_{j=1}^{k} \rho_i x_j - \sum_{j>k}^{i-1} \zeta_j x_j \\ \text{s.t.} \quad & f\left(\sum_{j<i} a_j x_j + a_i\right) \leq w \\ & \sum_{j<i} x_j \leq k-1,\ x_j \in \{0,1\}\ \forall\, j = 1, \ldots, i-1 \end{aligned} \tag{12}$$

From [15], We have the following result.

Lemma 2.2.1. *If the lifting coefficients ζ_j for $j = k+1, \ldots, i$ are computed as in* (12) *then the inequality*

$$f(0) + \sum_{j=1}^{k} \rho_j x_j + \sum_{j>k}^{i} \zeta_j x_j \leq w$$

is facet-defining for $\text{conv}(\mathcal{P}^i)$.

Lemma 2.2.2. *The lifting coefficient ζ_i in* (12) *can be computed in $O(i^3)$ time.*

Proof. The optimization problem (12) is equivalent to the following problem of minimizing concave function over cardinality constraint

$$\min \quad \left\{ f\left(\sum_{j<i} a_j x_j + a_i\right) - \sum_{j=1}^{k} \rho_j x_j - \sum_{j>k}^{i-1} \zeta_j x_j,\ \sum_{j<i} x_j \leq k-1,\ x_j \in \{0,1\}\ \forall\, j = 1, \ldots, i-1 \right\}.$$

This problem can be reduced to a two-dimension parametric linear programming problem and solved by enumerating all its $O(i^2)$ extreme points. Atamtürk and Narayanan give such an algorithm in [4] that finds an optimal solution in $O(i^3)$. □

The following theorem directly follows from Lemma 2.2.1 and Lemma 2.2.2.

Theorem 2.2.3. *For a permutation $((1), \ldots, (n))$, the inequality*

$$f(0) + \sum_{i \leq k} \rho_{(i)} x_{(i)} + \sum_{i>k} \zeta_{(i)} x_{(i)} \leq w \tag{13}$$

is facet-defining for $\text{conv}(\mathcal{P})$*, and can be computed in time $O(n^4)$.*

Next we relate the above inequality to the separation inequalities (10) when a has identical components.

Proposition 2.2.4. *If the components of a are identical, then $\zeta_i = f(k) - f(k-1)$ for $i > k$. Moreover for all $i > k$, the lifted coefficient $\zeta_i = \pi_i$, where π_i is given by* (7) *corresponding to $i_0 = k-1$.*

Proof. Denote $f(\sum_i a_i x_i)$ as $f(\sum_i x_i)$ and $f(a \cdot i)$ as $f(i)$. We have

$$\begin{aligned}
\zeta_i &= \min_{x \in \{0,1\}^i} \left\{ f\left(\sum_{j<i} x_j + 1\right) - f(0) - \sum_{j=1}^{k} \rho_j x_j - \sum_{j>k}^{i-1} \zeta_j x_j \;:\; \sum_{j<i} x_j \leq k-1 \right\} \\
&\geq \min_{x \in \{0,1\}^i} \left\{ f\left(\sum_{j<i} x_j + 1\right) - f\left(\sum_{j<i} x_j\right) \;:\; \sum_{j<i} x_j \leq k-1 \right\} \\
&= f(k) - f(k-1),
\end{aligned}$$

where the inequality comes from the validity of the coefficients ρ_j for $1 \leq j \leq k$ and ζ_j for $j > k$, i.e., $f(0) + \sum_{j=1}^{k} \rho_j x_j + \sum_{j>k}^{i-1} \zeta_j x_j \leq f(\sum_{j<i} x_j)$ and the equality comes from the concavity of f. However $\zeta_i \leq f(k) - f(k-1)$ since $x_j = 1$ for $j \in \{1, \ldots, k-1\}$ is a solution of (12). Finally we observe that $\zeta_i = \pi_i$ defined in (7) when $i_0 = k-1$ and $i > k$. □

The $O(n^4)$ time complexity of the exact lifted inequality (13) make it computationally ineffective in a branch-and-cut framework. Next, we derive another set of approximately lifted valid inequalities that will be used in our computational experiments.

Proposition 2.2.5. *For $i > k$, let $T_{(i)} = \operatorname{argmax} \{a(T) : T \subseteq \{(1), \ldots, (i-1)\}, |T| = k-1\}$, and $\gamma_{(i)} = f(a(T_{(i)}) + a_{(i)}) - f(a(T_{(i)}))$. The inequality,*

$$f(0) + \sum_{i \leq k} \rho_{(i)} x_{(i)} + \sum_{i>k} \gamma_{(i)} x_{(i)} \leq w \tag{14}$$

is valid for $\operatorname{conv}(\mathcal{P})$ *and can be computed in $O(n \log n)$ time.*

Proof. Without loss of generality, we fix the sequence as $(1, \ldots, n)$ in the proof. Let $T_i^* \subseteq \{1, \ldots, i-1\}$ denote the support of an optimal solution of (12), i.e. $x_j = 1$ for $j \in T_i^*$ is an optimal solution. We prove the statement by showing that for every $i > k$, $\gamma_i \leq \zeta_i$:

$$\begin{aligned}
\gamma_i &= f(a(T_i) + a_i) - f(a(T_i)) \\
&\leq f(a(T_i^*) + a_i) - f(a(T_i^*)) \\
&\leq f(a(T_i^*) + a_i) - f(0) - \sum_{j \in T_i^*, j \leq k} \rho_j - \sum_{j \in T_i^*, j > k} \zeta_j \\
&= \zeta_i.
\end{aligned}$$

Above the first inequality is by concavity of f and the fact that $a(T_i) \geq a(T_i^*)$. The second inequality comes from the validity of the coefficients ρ_j for $1 \leq j \leq k$ and ζ_j for $j > k$, i.e., $f(0) + \sum_{j \in T_i^*, j \leq k} \rho_j + \sum_{j \in T_i^*, j > k} \zeta_j \leq f(a(T_i^*))$. Finally the last equality is by definition of ζ_i.

For time complexity, the ρ_is can be computed in time $O(k)$. Then for γ_i, we maintain a sorted list of $a_1, \ldots, a_{i-1}$ and the sum of its $k-1$ largest items. For a new a_i, we insert it into the sorted list and update the sum, which takes $O(\log n)$ time. Therefore the total time will be $O(n \log n)$. □

We refer to the approximately lifted inequality (14) as a *lifted inequality* (LI) in the remainder of the paper.

Remark 2.2.6. If the components of a are identical then $\gamma_i = f(k) - f(k-1)$, and therefore $\gamma_i = \zeta_i$ in this case.

We also have the following property that the (LI) inequality is at least as strong as the (EP) inequality.

Proposition 2.2.7. *$\gamma_{(i)} - \rho_{(i)} \geq 0$, and is positive if f is strictly monotone.*

Proof. Since

$$\gamma_{(i)} - \rho_{(i)} = f(a(T_{(i)} + a_{(i)}) - f(a(T_{(i)}) - \left(f\left(\sum_{j<i} a_{(j)} + a_{(i)}\right) - f\left(\sum_{j<i} a_{(j)}\right)\right),$$

and we know from the definition in (14) that $a(T_{(i)}) < \sum_{j<i} a_{(j)}$ if $i > k$, therefore by concavity $\gamma_{(i)} \geq \rho_{(i)}$. It is then positive if f is strictly monotone. □

3 Computational Results

In this section we demonstrate the effectiveness of the proposed separation inequalities (10) and the lifted inequalities (14) for solving the following class of mean-risk knapsack problems [3]:

$$\min\left\{-\lambda' x + \Omega(\epsilon)\sqrt{\nu' x} : \; b'x \leq B, \; x \in \{0,1\}^n\right\}. \tag{15}$$

As discussed in Section 1, problem (15) involves a weighted combination of the mean and standard deviation of the total value of a knapsack, where individual item values are independent stochastic with mean μ_i and variance ν_i for $i = 1, \ldots, n$. Following [3, 6, 11] the tradeoff between the mean and standard deviation of the objective is set as $\Omega(\epsilon) = \sqrt{\frac{1-\epsilon}{\epsilon}}$ where $\epsilon \in (0,1)$ represents a risk aversion parameter. The vector of item weights and knapsack capacity are denoted by b and B, respectively. Problem (15) can be formulated as a mixed-integer second order cone program (SOCP):

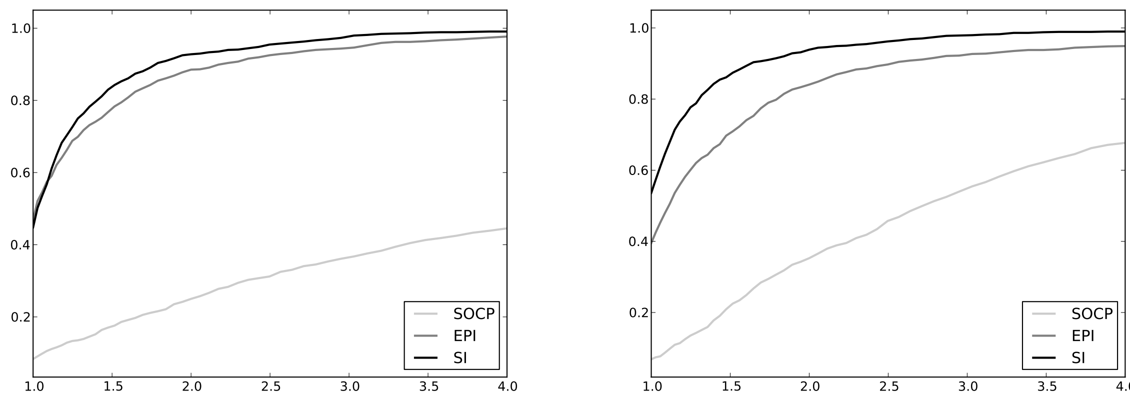
$$\min\left\{-\lambda' x + \Omega(\epsilon) w : \; w \geq \sqrt{\sum_i \nu_i x_i^2}, \; b'x \leq B, \; x \in \{0,1\}^n\right\}. \tag{16}$$

We incorporate the proposed inequalities in a branch-and-cut framework for the above formulation. Note that inequalities derived in the previous section are for the cardinality constraint $e'x \leq k$, while the constraint in (16) is a knapsack. We derive a cardinality constraint from the knapsack constraint as a simple cover inequality as follows. Sort b_i such that $b_1 \leq \cdots \leq b_n$, and find an index k such that $b_1 + \cdots b_k \leq B$ and $b_1 + \ldots + b_{k+1} > B$. Then the constraint $e'x \leq k$ is valid for $b'x \leq B$.

The implementation, written in python, is based on the mixed integer second order cone programming solver of Gurobi 5.6. The continuous relaxation of (16) at each node is solved using the Barrier method. We restrict the number of submodular inequalities added to at most five. Gurobi's internal cut parameters are in default setting. We disable multithreading, heuristics, and the concurrent MIP solver; and set the relative MIP optimality gap as 0.01% and the time limit of the computation to 30 minutes. All computations are on an Intel Xeon server with 16 cores and 7GB memory restriction.

Our computational experiments use randomly generated instances of (16). Our parameters' setting generally follow the ones in [3]. For the *weighted case*, each component of λ and b is generated from a uniform distribution with range $[0, 100]$. The risk aversion parameter ϵ is from $\{0.01, 0.02, 0.03\}$. It is varied to observe the relationship between the weight on the nonlinear term in the objective and running time. For each component ν_i of the variance vector, its square root is uniformly generated in the range $[0, \alpha\lambda_i]$ where α is from $\{0.5, 0.75, 1\}$. Such generation ensure that $\sqrt{\nu_i}/\lambda_i \leq \alpha$. Finally we set $B = \lfloor \sum_i b_i / r \rfloor$ where r is also a varied parameter. For a fixed $b_1, \ldots, b_n$, the larger r is, the smaller B is, thus the smaller k is. Therefore the parameter r controls the right-hand-side of the cardinality constraint. For the *unweighted case*, the uniform distribution of components in λ is changed to $[1, 5]$ and every component of ν_i is identical and equal to a number whose square root is generated according to the uniform distribution with range $[1, \alpha \cdot \min_i \{\lambda_i\}]$. The reasons for the changes are the following. First we need the upper bound $\min_i \{\lambda_i\}$ to ensure $\sqrt{\nu_i} \leq \alpha\lambda_i$ for every i. Second if we used a lower bound of 0 for λ, it sometimes generates very small λ_i which then makes the non-linear part of the problem negligible. Thirdly, if we used the original upper bound of 100 for λ, instances for the unweighted case could be solved in less than one second making them too trivial.

The experiments are performed over all combinations of the parameters n, α, ϵ and r; and for each combination we generate 20 instances. For each instance, we solve the problem using three different approaches: the SOCP formulation, SOCP formulation with the extended polymatroid inequalities (3) denoted by EPI,



(a) Weighed case (b) Unweighted

Figure 1: Performance Profile

and the SOCP formulation with lifted inequalities (14) denoted by LI (or separation inequalities (7) if unweighted, denoted by SI).

Figure 1 presents the performance profiles of time for the weighted and unweighted cases. Following Dolan and Moré in [7], the performance profiles are constructed as follows. We have a set $\mathcal{S}$ of three solvers: SOCP, EPI, LI/SI, and a set of problem instances $\mathcal{P}$. For a solver $s \in \mathcal{S}$ and a problem instance $p \in \mathcal{P}$, we calculate its performance ratio $r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}}$ where $t_{p,s}$ is the time required by solver s on instance p. Figure 1 plots the cumulative distribution function of the performance ratio defined as $g_s(\tau) = \frac{1}{|\mathcal{P}|} \left|\{p \in \mathcal{P} : r_{p,s} \leq \tau\}\right|$. Observe that in both cases the proposed inequalities are uniformly better than SOCP formulation and SOCP with EPI. The improvement in the unweighted cases is very significant since in this case we have a complete description of the convex hull.

In Tables 1 and 2, we provide more details of the performance improvement. These tables report, for both the weighted and unweighted cases, the node counts, the solution times, and the corresponding percentage of improvements over EPI across various parameter settings. We note that with LI or SI, performance is better in almost every parameter group. We also observe that the weighted case takes much more time and nodes than the unweighted case. When the parameter α , that control the range of the variance ν_i, gets smaller, the time and node needed decrease most significantly, compared with other parameters. This is because when α is small, the non-linear part of the constraint is less import. For both weighted and unweighted case, smaller α means that LI/SI cuts lead to better improvements. For ϵ, a smaller value puts more weight on the difficult nonlinear part of the objective and therefore the running time is longer. For the parameter r, we see that in the weighted case, larger r results in better improvement in case of the LI cuts. This observation matches our theoretical results since larger r indicates smaller k which implies that there will be more different coefficients in the LI cuts compared with EPI cuts.

	SOCP		EPI		LI			
	Nodes	Time	Nodes	Time	Nodes	Improvement	Time	Improvement
$n = 50$	4480	15.18	1416	8.17	1422	-0.46%	6.60	19.28%
$n = 100$	150638	662.04	26093	148.49	24209	7.22%	140.98	5.05%
$\alpha = 0.5$	17678	78.07	2145	20.10	2018	5.92%	12.03	40.16%
$\alpha = 0.75$	74241	335.07	12571	76.20	12412	1.27%	75.51	0.90%
$\alpha = 1$	140758	602.69	26547	138.68	24018	9.53%	133.83	3.50%
$\epsilon = 0.03$	39773	159.74	4041	23.84	3732	7.67%	19.01	20.27%
$\epsilon = 0.02$	75951	307.15	12245	70.27	11754	4.01%	64.93	7.60%
$\epsilon = 0.01$	116953	548.94	24977	140.88	22962	8.07%	137.44	2.44%
$r = 7.5$	64116	276.29	8004	55.48	8007	-0.03%	48.97	11.73%
$r = 5$	95961	420.91	12622	71.94	11194	11.31%	68.09	5.36%
$r = 2$	72599	318.63	20637	107.57	19247	6.74%	104.32	3.03%

Table 1: Performance Comparison for Weighted Case

	SOCP		EPI		SI			
	Nodes	Time	Nodes	Time	Nodes	Improvement	Time	Improvement
$n = 50$	3923	12.22	2962	8.92	2074	29.96%	7.85	12.02%
$n = 100$	21133	77.27	4248	22.10	3482	18.01%	16.27	26.41%
$\alpha = 0.5$	1643	5.04	773	5.55	731	5.33%	2.56	53.91%
$\alpha = 0.75$	6180	19.99	1600	5.95	1460	8.72%	5.51	7.47%
$\alpha = 1$	29761	109.20	8442	35.03	6144	27.22%	28.10	19.77%
$\epsilon = 0.03$	2287	7.78	905	3.43	838	7.35%	3.24	5.53%
$\epsilon = 0.02$	4631	15.31	1331	6.86	1262	5.17%	4.76	30.65%
$\epsilon = 0.01$	30666	111.13	8579	36.24	6235	27.32%	28.17	22.27%
$r = 7.5$	10300	35.35	4883	20.57	3153	35.43%	12.67	38.42%
$r = 5$	13785	52.23	4613	17.74	3871	16.09%	18.44	-3.96%
$r = 2$	13498	46.65	1319	8.22	1312	0.52%	5.06	38.47%

Table 2: Performance Comparison for Unweighted Case

Acknowledgement

This research has been supported in part by the National Science Foundation (Grant 1129871) and the Air Force Office of Scientific Research (Grant FA9550-12-1-0154).

References

[1] S. AHMED AND A. ATAMTÜRK, *Maximizing a class of submodular utility functions*, Mathematical programming, 128 (2011), pp. 149–169.

[2] S. AHMED AND D. J. PAPAGEORGIOU, *Probabilistic set covering with correlations*, Operations Research, 61 (2013), pp. 438–452.

[3] A. ATAMTÜRK AND V. NARAYANAN, *Polymatroids and mean-risk minimization in discrete optimization*, Operations Research Letters, 36 (2008), pp. 618–622.

[4] A. ATAMTÜRK AND V. NARAYANAN, *The submodular knapsack polytope*, Discrete Optimization, 6 (2009), pp. 333 – 344.

[5] F. BAUMANN, S. BERCKEY, AND C. BUCHHEIM, *Exact algorithms for combinatorial optimization problems with submodular objective functions*, in Facets of Combinatorial Optimization, M. Jünger and G. Reinelt, eds., Springer Berlin Heidelberg, 2013, pp. 271–294.

[6] D. BERTSIMAS AND I. POPESCU, *Optimal inequalities in probability theory: A convex optimization approach*, SIAM Journal on Optimization, 15 (2005), pp. 780–804.

[7] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.

[8] J. EDMONDS, *Submodular functions, matroids, and certain polyhedra*, in Combinatorial Structures and Their Applications, Gordon and Breach, 1971, pp. 68–87.

[9] E. FELDMAN, F. LEHRER, AND T. RAY, *Warehouse location under continuous economies of scale*, Management Science, 12 (1966), pp. 670–684.

[10] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

[11] L. E. GHAOUI, M. OKS, AND F. OUSTRY, *Worst-case value-at-risk and robust portfolio optimization: A conic programming approach*, Operations Research, 51 (2003), pp. 543–556.

[12] G. GOEL, C. KARANDE, P. TRIPATHI, AND L. WANG, *Approximability of combinatorial problems with multi-agent submodular cost functions*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS, IEEE, 2009, pp. 755–764.

[13] M. X. GOEMANS, N. J. A. HARVEY, S. IWATA, AND V. MIRROKNI, *Approximating submodular functions everywhere*, in SODA '09: Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, 2009, Society for Industrial and Applied Mathematics, pp. 535–544.

[14] M. X. GOEMANS AND J. A. SOTO, *Symmetric submodular function minimization under hereditary family constraints*, CoRR, abs/1007.2140 (2010).

[15] Z. GU, G. L. NEMHAUSER, AND M. W. SAVELSBERGH, *Sequence independent lifting in mixed integer programming*, Journal of Combinatorial Optimization, 4 (2000), pp. 109–129.

[16] M. HAJIAGHAYI, M. MAHDIAN, AND V. MIRROKNI, *The facility location problem with general cost functions*, Networks, 42 (2003), pp. 42–47.

[17] R. HASSIN AND A. TAMIR, *Maximizing classes of two-parameter objectives over matroids*, Mathematics of Operations Research, 14 (1989), pp. 362–375.

[18] H. ISHII, S. SHIODE, T. NISHIDA, AND Y. NAMASUYA, *Stochastic spanning tree problem*, Discrete Applied Mathematics, 3 (1981), pp. 263 – 273. Special Copy.

[19] S. IWATA AND K. NAGANO, *Submodular function minimization under covering constraints*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS, IEEE, 2009, pp. 671–680.

[20] L. LOVÁSZ, *Submodular functions and convexity*, in Mathematical Programming The State of the Art, Springer, 1983, pp. 235–257.

[21] E. NIKOLOVA, *Approximation algorithms for reliable stochastic combinatorial optimization*, in Proceedings of the 13th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX, Springer, 2010, pp. 338–351.

[22] S. ONN AND U. G. ROTHBLUM, *Convex combinatorial optimization*, Discrete & Computational Geometry, 32 (2004), pp. 549–566.

[23] L. OZSEN, C. R. COULLARD, AND M. S. DASKIN, *Capacitated warehouse location model with risk pooling*, Naval Research Logistics (NRL), 55 (2008), pp. 295–312.

[24] H. E. ROMEIJN, T. C. SHARKEY, Z.-J. M. SHEN, AND J. ZHANG, *Integrating facility location and production planning decisions*, Networks, 55 (2010), pp. 78–89.

[25] D. STRATILA, *Combinatorial Optimization Problems with Concave Costs*, PhD thesis, MIT, 2009.

[26] Z. SVITKINA AND L. FLEISCHER, *Submodular approximation: Sampling-based algorithms and lower bounds*, SIAM Journal on Computing, 40 (2011), pp. 1715–1737.

[27] D. M. TOPKIS, *Supermodularity and complementarity*, Princeton University Press, 1998.

Appendix: Proof of Lemma 2.1.3

Lemma 2.1.3. *Suppose i_0 is constructed as in* (6) *then the following linear system in $P(S)$ for $S \in \mathcal{S}$*

$$\sum_{S:i\in S} P(\{1,\dots,i_0\}\cup S) = x_i,\ i \in \{i_0+1,\dots,n\}$$

has a nonnegative solution and

$$\sum_{S\in\mathcal{S}} P(\{1,\dots,i_0\}\cup S) = \frac{\sum_{i=i_0+1}^{n} x_i}{k-i_0}.$$

Proof. To avoid confusion with the original notation, we change the notation as follows. Set $m = n - i_0, l = k - i_0$, $v_1 = x_{i_0+1}, v_2 = x_{i_0+2}, \dots, v_m = x_n$. We will prove for $m, l \in \mathbb{Z}_+, m \geq l+1$, $v_1 \geq v_2 \geq \cdots v_m \geq 0$, and $\mathcal{S} = \{S : S \subset \{1,\dots,m\}, |S| = l\}$, the following linear system in $q(S), S \in \mathcal{S}$

$$\sum_{S:j\in S} q(S) = v_j, j \in \{1,\dots,m\},$$

has a nonnegative solution and $\sum_{S\in\mathcal{S}} q(S) = \frac{\sum_{j=1}^{m} v_j}{l}$.

We will use Algorithm 1 to construct a feasible solution. The main idea is for any set S, the associated variable $q^t(S)$ will remain nondecreasing in the iteration count t. For any j, v_j^t the right-hand-side value that has not been satisfied yet, and it will remain nonincreasing in t. We will keep $v_j^{t+1} + \sum_{S,j\in S,S\in\mathcal{S}} q^{t+1}(S) = v_j$ along the way. At the end of the procedure, every item j will have its $v_j^t = 0$, and we find the solution as desired.

We now exploit some properties of v_j^t and $q^t(S)$. Initially, we have

Algorithm 1: Recursive procedure

$t \leftarrow 0, m^t \leftarrow m;$

$v_j^t \leftarrow v_j, 1 \le j \le m^t \quad q^t(S) \leftarrow 0, \forall S \in \mathcal{S};$

while $m^t > l+1$ **do**

C1   **if** $lv_l^t \le \sum_{j=1}^{m^t} v_j^t - lv_{m^t}^t$ **then**

$\quad q^{t+1}(\{1,\dots,l-1,m^t\}) \leftarrow q(\{1,\dots,l-1,m^t\}) + v_{m^t}^t;$

$\quad v_j^{t+1} \leftarrow v_j^t - v_{m^t}^t, j \in \{1,\dots,l-1\};$

$\quad m^{t+1} \leftarrow m^t - 1;$

else

C2 $\quad \Delta = \sum_{j=1}^{m^t} v_j^t / l - v_{l+1}^t;$

$\quad q^{t+1}(\{1,\dots,l\}) \leftarrow q^t(\{1,\dots,l\}) + \Delta;$

$\quad v_j^{t+1} \leftarrow v_j^t - \Delta, j \in \{1,\dots,l\};$

$\quad m^{t+1} \leftarrow m^t$

$q^{t+1}(S) \leftarrow q^t(S)$ for all the other S;

SRT   sort $v_1^{t+1},\dots,v_{m^t}^{t+1}$ and index them so that $v_1^{t+1} \ \ge \cdots v_{m^{t+1}}^{t+1};$

$t \leftarrow t+1;$

$q^t(\{1,\dots,l+1\} \setminus \{j\}) \leftarrow q^t(\{1,\dots,l+1\} \setminus \{j\}) + \frac{\sum_{\tau=1}^{l+1} v_\tau^t}{l} - v_j^t, j \in \{1,\dots,l+1\}$

1. $v_j^0 \ge 0$ for every j
2. $q^0(S) = 0$ and $v_j^0 + \sum_{S, j\in S, S\in\mathcal{S}} q^0(S) = v^j$
3. Since $y \ge z_{i_0+1}$, we have $\sum_{j=k}^n x_j \ge (k - i_0 - 1)x_{i_0+1} - \sum_{j=i_0+2}^{k-1} x_j$. Then $\sum_{j=1}^m v_j = \sum_{j=i_0+1}^n x_j \ge (k-i_0)x_{i_0+1} = lv_1$.

We will show that v_ls satisfy the following invariance after finishing line SRT.

1. $v_j^{t+1} \ge 0$ for $j \in \{1,\dots,m^{t+1}\}$.
2. $v_j^{t+1} + \sum_{S,j\in S,S\in\mathcal{S}} q^{t+1}(S) = v_j$.
3. $lv_1^{t+1} \le \sum_{j=1}^{m^{t+1}} v_j^{t+1}$.

First we prove $v_j^{t+1} \ge 0$. Case C1 is easy since $v_j^t \ge v_{m^t}^t$ for all j. For Case C2 where $lv_l^t > \sum_{j=1}^{m^t} v_j^t - lv_{m^t}^t$, the smallest v_j^{t+1} among the ones updated is

$$v_l^t - \Delta = v_l^t - \frac{\sum_{j=1}^{m^t} v_j^t}{l} + v_{l+1}^t \ge \frac{\sum_{j=1}^{m^t} v_j^t}{l} - v_{m^t}^t - \frac{\sum_{j=1}^{m^t} v_j}{l} + v_{l+1}^t = v_{l+1}^t - v_{m^t}^t \ge 0.$$

Then we show $v_j^{t+1} + \sum_{S,j\in S,S\in\mathcal{S}} q^{t+1}(S) = v_j$. Since in the loop each iteration we only consider one set S, and for any item $j \in S$, $v_j^t - v_j^{t+1} = q^{t+1}(S) - q^t(S)$, the claim is true.

Then we prove that $lv_1^{t+1} \le \sum_{j=1}^{m^{t+1}} v_j^{t+1}$ after line SRT. In Case C1, v_1^{t+1} will be either

- $v_1^t - v_{m^t}^t$. Then $\sum_{j=1}^{m^{t+1}} v_j^{t+1} = \sum_{j=1}^{m^t} v_j^t - lv_{m^t}^t \ge lv_1^t - lv_{m^t}^t$ since the claim holds for the previous iteration, or
- v_l^t. Then again $\sum_{j=1}^{m^{t+1}} v_j^{t+1} = \sum_{j=1}^{m^t} v_j^t - lv_{m^t}^t \ge lv_l^t$ because we are at Case C1.

In the second case C2, first notice that after line SRT, $v_1^{t+1} = v_{l+1}^t$ since the other choice $v_1^t - \Delta = v_1^t - \frac{\sum_{j=1}^{m^t} v_j^t}{l} + v_{l+1}^t \le v_{l+1}^t$ because $l v_1^t \le \sum_{j=1}^{m^t} v_j^t$. Then

$$\sum_{j=1}^{m^{t+1}} v_j^{t+1} = \sum_{j=1}^{m^t} v_j^t - l\Delta = \sum_{j=1}^{m^t} v_j^t - l\left(\frac{\sum_{j=1}^{m^t} v_j^t}{l} - v_{l+1}^t\right) = l v_{l+1}^t.$$

If Algorithm 1 terminates, we say that $q^t(S), \forall S \in \mathcal{S}$ is the solution desired. We consider two cases after the while loop. For any $j > l+1$, we have $v_j^t = 0$ which is equivalent to $\sum_{S, j\in S, S\in\mathcal{S}} q^t(S) = v_j$. For any $j \le l+1$, before the last line, we have $\sum_{S,j\in S,S\in\mathcal{S}} q^t(S) = v_j - v_j^t$, and after last line, we have

$$\sum_{S,j\in S,S\in\mathcal{S}} q^t(S) = v_j - v_j^t + \sum_{j'\neq j}\left(\frac{\sum_{\tau=1}^{l+1} v_\tau^t}{l} - v_{j'}^t\right) = v_j - v_j^t + \sum_{\tau=1}^{l+1} v_\tau^t - \sum_{j'\neq j} v_{j'}^t = v_j.$$

If it does not terminate, we claim that $q(S) = \lim_{t\to\infty} q^t(S)$ for all $S \in \mathcal{S}$ is the solution desired. To prove that, we show that $\lim_{t\to\infty} v_j^t = 0$. If this is true, then we get $\lim_{t\to\infty} \sum_{S,j\in S,S\in\mathcal{S}} q^t(S) = v_j$. Notice that each time after we finish Case (C2), $\sum_{j=1}^{m^{t+1}} v_j^{t+1} \le \frac{l}{l+1}\sum_{j=1}^{m^t} v_j^t$ because

$$\sum_{j=1}^{m^t} v_j^t \ge \sum_{j=1}^{l+1} v_j^t \ge (l+1)v_{l+1}^t, \text{ and } \sum_{j=1}^{m^{t+1}} v_j^{t+1} = \sum_{j=1}^{m^t} v_j^t - l\Delta = l v_{l+1}.$$

Since the algorithm does not terminate, Case C2 happens infinitely many times. Notice that initially $\sum_{j=1}^{m^0} v_j \le m$, the value

$$\lim_{t\to\infty}\sum_{j=1}^{m^t} v_j^t \le \lim_{t\to\infty}\left(\frac{l}{l+1}\right)^t m = 0.$$

Since v_j^t is always nonnegative, $\lim_{t\to\infty}\sum_{j=1}^{m^t} v_j^t = 0$ and $\lim_{t\to\infty} v_j^t = 0$.

Now we calculate the sum of all variables. Given that

$$\sum_{S:j\in S} q(S) = v_j, j \in \{1,\ldots,m\},$$

we have

$$\sum_{j=1}^m \sum_{S:j\in S} q(S) = \sum_{j=1}^m v_j.$$

Since each $q(S)$ appears on the left side exactly l times,

$$l\sum_{S\in\mathcal{S}} q(S) = \sum_{j=1}^m v_j.$$

□