Discrete Optimization ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

The submodular knapsack polytope

Alper Atamtürk [a,*], Vishnu Narayanan [b]

[a] Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720–1777, USA

[b] Industrial Engineering and Operations Research, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

ARTICLE INFO

Article history:
Received 21 July 2008
Received in revised form 17 February 2009
Accepted 31 March 2009
Available online xxxx

Keywords:
Conic integer programming
Submodular functions
Probabilistic knapsacks
Branch-and-cut algorithms

ABSTRACT

The submodular knapsack set is the discrete lower level set of a submodular function. The modular case reduces to the classical linear 0–1 knapsack set. One motivation for studying the submodular knapsack polytope is to address 0–1 programming problems with uncertain coefficients. Under various assumptions, a probabilistic constraint on 0–1 variables can be modeled as a submodular knapsack set.

In this paper we describe cover inequalities for the submodular knapsack set and investigate their lifting problem. Each lifting problem is itself an optimization problem over a submodular knapsack set. We give sequence-independent upper and lower bounds on the valid lifting coefficients and show that whereas the upper bound can be computed in polynomial time, the lower bound problem is $\mathcal{NP}$-hard. Furthermore, we present polynomial algorithms based on parametric linear programming and computational results for the conic quadratic 0–1 knapsack case.

1. Introduction

We consider the polytope defined by the convex hull of the discrete lower level set of a submodular set function. Given a finite ground set N, a set function $f : 2^N \to \mathbb{R}$ is submodular on N if

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \quad \text{for all } S, T \subseteq N.$$

Throughout, by abusing notation, we refer to a set function also as $f(\chi_S)$, where χ_S denotes the binary characteristic vector of $S \subseteq N$. Given a submodular function f on N and $b \in \mathbb{R}$, *the submodular knapsack set* is

$$X := \left\{x \in \{0, 1\}^N : f(x) \leq b\right\}.$$

For modular f, X reduces to the classical linear 0–1 knapsack set.

For notational simplicity, we denote a singleton set $\{i\}$ with its unique element i. For a vector $v \in \mathbb{R}^N$, we use $v(S)$ to denote $\sum_{i\in S} v_i$ for $S \subseteq N$. Given a set function f on N and $i \in N$, let the *difference function* be

$$\rho_i(S) := f(S \cup i) - f(S) \quad \text{for } S \subseteq N \setminus i.$$

It is easy to check that f is submodular if and only if $\rho_i(S) \geq \rho_i(T)$ for all $S \subseteq T \subseteq N \setminus i$ and $i \in N$; that is, the difference function ρ_i is nonincreasing on $N \setminus i$ (see e.g. [1]).

* Corresponding address: Department of Industrial Engineering and Operations Research, University of California, 4141 Etcheverry Hall MC 1777, Berkeley, CA 94720–1777, USA. Tel.: +1 510 642 4559; fax: +1 510 642 1403.

E-mail addresses: atamturk@berkeley.edu (A. Atamtürk), vishnu@iitb.ac.in (V. Narayanan).

1572-5286/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2009.03.002

Motivation. Our motivation for studying the submodular knapsack polytope is to address linear 0–1 knapsack problems with uncertain coefficients. If the knapsack coefficients $\tilde{a}_i, i \in N$, are random variables, then for small $\epsilon > 0$ a probabilistic (chance) constraint [2]

$$\textbf{Prob}(\tilde{a}x \le b) \ge 1 - \epsilon \tag{1}$$

on $x \in \{0,1\}^N$ can be modeled as a conic quadratic 0–1 knapsack set

$$X_{CQ} := \left\{ x \in \{0,1\}^N : ax + \Omega(\epsilon)\|Dx\| \le b \right\},$$

where a_i is a nominal value and d_i is a deviation statistic for $\tilde{a}_i$, $i \in N$, $D = \text{diag}(d_1, d_2, \ldots, d_{|N|})$, $\Omega(\epsilon) > 0$, and $\|\cdot\|$ is the L2 norm. The term $\Omega(\epsilon)\|Dx\|$ is used to build sufficient slack into the constraint to accommodate the variability of $\tilde{a}$ around the nominal value a. If $\tilde{a}_i$'s are normally distributed independent random variables, then letting a_i and d_i be the mean and standard deviation of $\tilde{a}_i$, $i \in N$, and $\Omega(\epsilon) = -\Phi^{-1}(\epsilon)$ with $0 < \epsilon \le 0.5$, where Φ is the standard normal cumulative distribution function, the set of 0–1 solutions for the probabilistic knapsack constraint (1) is exactly X_{CQ} (see e.g [3]). On the other hand, if $\tilde{a}_i$'s are known only through their first two moments a_i and d_i^2, then any point in X_{CQ} with $\Omega(\epsilon) = \sqrt{(1-\epsilon)/\epsilon}$ satisfies the probabilistic constraint (1) [4,5]. Alternatively, if $\tilde{a}_i$'s are only known to be symmetric with support $[a_i - d_i, a_i + d_i]$, then points in X_{CQ} with $\Omega(\epsilon) = \sqrt{\ln(1/\epsilon)}$ satisfy constraint (1) [6,7]. Hence, under different assumptions of uncertainty on $\tilde{a}$, one arrives at different instances of the conic quadratic knapsack set X_{CQ}.

Consider now a set function $f : 2^N \to \mathbb{R}$ defined as

$$f(S) = a(S) + g(c(S)), \tag{2}$$

where g is a concave function and $a(S)$ and $c(S)$ denote the sums of the components of $a, c \in \mathbb{R}^N$ on $S \subseteq N$. It is easy to check that if $c \ge \mathbf{0}$, then f is submodular on N (see e.g. [8]). To see that X_{CQ} is a special case of X, observe that the conic quadratic constraint defining X_{CQ} can be written as

$$ax + \Omega\sqrt{x'D^2x} \le b. \tag{3}$$

Because x is a binary vector, letting $c_i = \Omega^2 d_i^2$ for $i \in N$, we see that χ_S, $S \subseteq N$ is feasible for (3) if and only if S satisfies $f(S) = a(S) + \sqrt{c(S)} \le b$.

Although the polyhedral results in this paper are for the more general submodular knapsack polytope conv(X), we give efficient algorithms for a set function of the form (2). Because X_{CQ} reduces to the linear 0–1 knapsack set when $D = \mathbf{0}$, optimization over X is $\mathcal{NP}$-hard. Also as $X \subseteq \{0,1\}^N$, conv(X) is a polyhedral set.

Relevant literature. Most of the literature on the knapsack problem is for the linear case [9]. The polyhedral analysis of the linear knapsack set was initiated by Balas [10], Hammer et al. [11], and Wolsey [12]. See also [13–15]. For a recent review of the polyhedral results on the linear knapsack set we refer the reader to [16]. The majority of the research on the nonlinear knapsack problem is devoted to the case with separable nonlinear functions; see e.g. [17–19]. There are few studies on the nonseparable knapsack problem, most notably on the knapsack problem with quadratic objective and linear constraint [20,21]. Helmberg et al. [22] give SDP relaxations of knapsack problems with quadratic objective. Gallo and Simeone [23] give a Lagrangian approach for maximizing a supermodular function over a linear knapsack constraint. Sviridenko [24] gives an approximation algorithm for maximizing a submodular function over a linear knapsack constraint. Atamtürk and Narayanan [25] give a cutting plane approach for minimizing a submodular function of the form (2) over a discrete set. Ahmed and Atamtürk [8] consider maximizing the same function. We refer the reader to [26] for a survey of nonlinear knapsack problems. General mixed-integer rounding and disjunctive approaches have been developed by Atamtürk and Narayanan [27] Çezik and Iyengar [28] for conic quadratic mixed integer sets, but these approaches do not exploit any special structure. It appears that the submodular knapsack set X has not been considered in the literature before.

Outline. In Section 2 we describe linear inequalities for X. In particular, we present cover inequalities for X and discuss the lifting problem associated with them. The lifting problems of the cover inequalities for X are themselves optimization problems over submodular knapsack sets. We give upper and lower bounds on valid lifting coefficients. In Section 3 we give efficient algorithms for computing the bounds, approximations of the lifting coefficients and for separation for the cover inequalities for the set function of the form (2). In Section 4 we present computational results on the conic quadratic case.

2. Polyhedral analysis

In this section we describe valid inequalities for X. In particular, we discuss cover inequalities for X and their lifting. Throughout the section we make the following assumptions:

($\mathcal{A}$.1) f is nondecreasing,
($\mathcal{A}$.2) $f(\emptyset) = 0$,
($\mathcal{A}$.3) $0 < \rho_i(\emptyset) \le b$ for all $i \in N$.

Because f is submodular, assumption ($\mathcal{A}$.1) is equivalent to $\rho_i(N \setminus i) \ge 0$ for all $i \in N$, which can be checked easily. Assumption ($\mathcal{A}$.1) holds, for instance, for a function f of the form (2) if $a \ge \mathbf{0}$. If f is nondecreasing, then X is an independence system; that is, $\chi_T \in X$ implies $\chi_S \in X$ for all $S \subseteq T \subseteq N$. Assumption ($\mathcal{A}$.2) can be made without loss of generality as f can be translated otherwise. If ($\mathcal{A}$.1) and ($\mathcal{A}$.2) hold, assumption ($\mathcal{A}$.3) can be made without loss of generality. Because f is nondecreasing, we have $\rho_i(\emptyset) \geq 0$. However, submodularity and $\rho_i(\emptyset) = 0$ implies that $\rho_i(S) = 0$ for all $S \subseteq N \setminus i$, in which case x_i can be removed from X. Finally, if $\rho_i(\emptyset) > b$, then $x_i = 0$ in every feasible solution.

2.1. Valid inequalities

It is easy to see from ($\mathcal{A}$.2) and ($\mathcal{A}$.3) that conv(X) is a full-dimensional polytope. The following results on independence systems are standard and easy to verify (see e.g. [11]).

Proposition 1. *Inequality $x_i \geq 0,\ i \in N$, is facet-defining for* conv(X).

Proposition 2. *Inequality $x_i \leq 1,\ i \in N$, is facet-defining for* conv(X) *if and only if $f(\{i, j\}) \leq b$ for all $j \in N \setminus i$.*

Proposition 3. *If $\alpha x \leq \beta$ defines a facet of* conv(X) *not including* **0**, *then $\alpha \geq 0$ and $\beta > 0$.*

Definition 1. A subset S of N is said to be a *cover* for X if $\lambda := f(S) - b > 0$. A cover S is *minimal* if $f(S \setminus i) \leq b$ for all $i \in S$.

For a cover $S \subseteq N$ for X let the *cover inequality* be

$$x(S) \leq |S| - 1. \tag{4}$$

The cover inequality simply states that not all elements in a cover can be picked simultaneously to satisfy the knapsack constraint $f(x) \leq b$. Let

$$X(S) := \{x \in X : x_i = 0 \quad \text{for } i \in N \setminus S\}.$$

Proposition 4. *If $S \subseteq N$ is a cover for X, then cover inequality (4) is valid for X. Moreover, (4) defines a facet of* conv($X(S)$) *if and only if S is a minimal cover.*

The cover inequalities, typically, do not define facets of conv(X); however, they can be strengthened by extending them with non-cover elements. Unlike the linear case, for the submodular knapsack set, even the simple extensions are sequence-dependent. Proposition 5 describes such an extension of the cover inequalities (4).

Definition 2. Let $\pi = (\pi_1, \ldots, \pi_{|N \setminus S|})$ be a permutation of the elements of $N \setminus S$ and $S_i = S \cup \{\pi_1, \ldots \pi_i\}$ for $i = 1, \ldots, |N \setminus S|$ with $S_0 = S$. The *extension* of $S \subseteq N$ with respect to π is defined as $E_\pi(S) := S \cup U_\pi(S)$, where

$$U_\pi(S) := \left\{\pi_j \in N \setminus S : \rho_{\pi_j}(S_{j-1}) \geq \rho_i(\emptyset) \text{ for all } i \in S\right\}.$$

Proposition 5. *If $S \subseteq N$ is a cover for X, the extended cover inequality*

$$x(E_\pi(S)) \leq |S| - 1 \tag{5}$$

is valid for X. Moreover, inequality (5) defines a facet of conv($X(E_\pi(S))$) *if S is a minimal cover and for each $i \in U_\pi(S)$ there exist distinct $j_i, k_i \in S$ such that $f(S \cup \{i\} \setminus \{j_i, k_i\}) \leq b$.*

Proof. Let T be a subset of $E_\pi(S)$ of cardinality at least $|S|$. It is sufficient to show that $f(T) > b$. Let $K = S \setminus T$ and $L = U_\pi(S) \cap T =: \left\{\ell_1, \ldots, \ell_{|L|}\right\}$ indexed consistently with permutation π. Because $T = S \cup L \setminus K$ and $|T| \geq |S|$, we have $|K| \leq |L|$. Now using submodularity of f,

$$\begin{aligned} f(T) &= f(S \setminus K) + \sum_{\ell_i \in L} \rho_{\ell_i}(S \cup \{\ell_1, \ldots, \ell_{i-1}\} \setminus K) \\ &\geq f(S \setminus K) + \sum_{\pi_j \in L} \rho_{\pi_j}(S_{j-1}) \\ &\geq f(S \setminus K) + \sum_{i \in K} \rho_i(\emptyset) \\ &\geq f(S \setminus K) + \sum_{i \in K} \rho_i(S \setminus K) \geq f(S) > b. \end{aligned}$$

The second inequality follows from the definition of $U_\pi(S)$, $|K| \leq |L|$, and assumption ($\mathcal{A}$.1).

For the second part of the proposition, it is easy to see that the points $\chi_{S \setminus i}$ for all $i \in S$ and $\chi_{S \cup i \setminus \{j_i, k_i\}}$ for all $i \in U_\pi(S)$ and $j_i, k_i \in S$ are affinely independent and are on the face defined by (5). □

Example 1. Consider the conic quadratic knapsack set

$$\widehat{X}_{CQ} = \left\{x \in \{0, 1\}^5 : 2x_1 + 2x_2 + 2x_3 + x_4 + x_5 + \sqrt{x_4 + x_5} \leq 5.5\right\}.$$

For $S = \{1, 2, 3\}$ we have $\lambda = f(S) - 5.5 = 0.5 > 0$ and the corresponding minimal cover inequality

$$x_1 + x_2 + x_3 \leq 2.$$

For permutation $(4, 5)$, $S_1 = \{1, 2, 3, 4\}$ and $S_2 = \{1, 2, 3, 4, 5\}$. As $\rho_4(S_0) = 1 + \sqrt{1} = 2$ and $\rho_5(S_1) = 1 + \sqrt{2} - \sqrt{1} = 1.41$, the corresponding extension $E_{(4,5)}(S) = \{1, 2, 3, 4\}$ gives the extended cover inequality

$$x_1 + x_2 + x_3 + x_4 \leq 2.$$

Similarly, for permutation $(5, 4)$, we have extension $E_{(5,4)}(S) = \{1, 2, 3, 5\}$ giving the extended cover inequality

$$x_1 + x_2 + x_3 + x_5 \leq 2.$$

It is easily checked that both inequalities indeed define facets of $\text{conv}(\widehat{X}_{CQ})$.

2.2. *Lifting cover inequalities for submodular knapsacks*

In this section we study the lifting problem of the cover inequalities in order to strengthen them. The lifting procedure has been very effective in strengthening inequalities for the linear 0–1 knapsack set (see [10,29,30,14,11,12] among others). The lifting problem for cover inequalities for X is itself an optimization problem over the submodular knapsack set.

Precisely, we lift the cover inequality (4) to a valid inequality of the form

$$x(S) + \sum_{i \in N \setminus S} \alpha_i x_i \leq |S| - 1. \tag{6}$$

The lifting coefficients α_i, $i \in N \setminus S$ can be computed iteratively in some sequence: Suppose the cover inequality (4) is lifted with variables x_i, $i \in J \subseteq N \setminus S$ to obtain the intermediate valid inequality

$$x(S) + \sum_{i \in J} \alpha_i x_i \leq |S| - 1 \tag{7}$$

in some sequence of J. Then x_k, $k \in N \setminus I$, where $I = S \cup J$ can be introduced to (7) by computing

$$\alpha_k = |S| - 1 - \varphi(I, k), \tag{8}$$

where

$$\varphi(I, k) := \max_{T \subseteq I} \left\{ |S \cap T| + \sum_{i \in J \cap T} \alpha_i : f(T \cup k) \leq b \right\}. \tag{9}$$

The lifting coefficients are typically a function of the sequence used for lifting. The extension given in Proposition 5 may be seen as a simple approximation of the lifted inequalities (7).

Proposition 6. *If S is a minimal cover for X, for any lifting sequence inequality* (6) *with coefficients satisfying* (8) *is facet-defining for* $\text{conv}(X)$.

For a deeper understanding of the structure of the lifted inequalities, it is of interest to identify bounds on the lifting coefficients that are independent of a chosen lifting sequence. We start with a simple lemma.

Lemma 1. *Let $S \subseteq N$ be a minimal cover and for $h = 0, \ldots, |S|$ let*

$$\mu_h := \max \{f(T) : |T| = h,\ T \subseteq S\}, \tag{10}$$

$$\nu_h := \min \{f(T) : |T| = h,\ T \subseteq S\}. \tag{11}$$

Then, for all $h = 0, \ldots, |S| - 1$ the following inequalities hold:

(i) $\mu_h \leq \mu_{h+1} - \lambda$, *and* (ii) $\nu_h \leq \nu_{h+1} - \lambda$.

Proof. Because S is a minimal cover, we have $\rho_i(S \setminus i) \geq \lambda$ for all $i \in S$.

(i) Let T_h^* be an optimal solution for (10) and $k \in S \setminus T_h^*$. It follows from submodularity of f and minimality of cover S that

$$\mu_h = f(T_h^*) \leq f(T_h^* \cup k) - \lambda \leq \mu_{h+1} - \lambda.$$

(ii) For this part, let ν_{h+1} be given by T_{h+1}^* and $k \in T_{h+1}^*$. Then by submodularity of f and minimality of cover S we have

$$\nu_{h+1} - \lambda = f(T_{h+1}^*) - \lambda \geq f(T_{h+1}^* \setminus k) \geq \nu_h. \quad \square$$

Please cite this article in press as: A. Atamtürk, V. Narayanan, The submodular knapsack polytope, Discrete Optimization (2009), doi:10.1016/j.disopt.2009.03.002

Table 1
Bounds for $\rho_5(\emptyset)$ and $\rho_5(N \setminus 5)$.

h	μ_h	ν_h	$b - \nu_{3-h}$
1	2.21	1.50	3.59
2	4.21	2.91	5.00
3	5.72	4.58	6.50
4	6.58	6.58	–

Proposition 7. *Let $S \subseteq N$ be a cover with $\lambda := f(S) - b > 0$ and μ_h and ν_h, $h = 0, \ldots, |S|$ be defined as in Lemma 1. Suppose that the lifted cover inequality*

$$x(S) + \sum_{i \in N \setminus S} \alpha_i x_i \leq |S| - 1 \tag{12}$$

defines a facet of conv(X)*. For any $i \in N \setminus S$, the following statements hold:*
1. *If $\rho_i(N \setminus i) \geq \mu_h$, then $\alpha_i \geq h$;*
2. *If $\rho_i(\emptyset) \leq b - \nu_{|S|-1-h}$, then $\alpha_i \leq h$.*

Proof. 1. The lifting coefficient of x_i, $i \in N \setminus S$, is smallest if x_i the last variable introduced to (12) in a lifting sequence. So, let $\alpha_i = |S| - 1 - \varphi(N \setminus i, i)$. Also, because the intermediate lifting inequality before introducing x_i is valid for X, we have $|S| - 1 \geq \varphi_{N \setminus i}(0, 0)$. Therefore, it is sufficient to show that $\varphi(N \setminus i, \emptyset) - \varphi(N \setminus i, i) \geq h$.

We claim that in any feasible solution to the lifting problem for x_i (when it is lifted last), no more than $|S| - 1 - h$ variables in S are positive. For contradiction, suppose at least $|S| - h$ variables in S are positive. Let S' denote the set of positive variables. Then, $|S \setminus S'| \leq h$; and by submodularity and our assumption, we have that $f(S') \geq f(S) - f(S \setminus S') > b - \mu_h \geq b - \rho_i(N \setminus i)$. Hence, we see that

$$f(S' \cup i) = f(S') + \rho_i(S' \setminus i) \geq f(S') + \rho_i(N \setminus i) > b.$$

Thus $\chi_{S'}$ is infeasible for $L(N \setminus i, i)$.

Now let S^* be an optimal solution set to this lifting problem. Let $\widehat{S} \subseteq S$ be such that $|\widehat{S}| = h$, and $\widehat{S} \cap S^* = \emptyset$. By the argument in the preceding paragraph such $\widehat{S}$ exists. Then, we claim that $S^* \cup \widehat{S}$ is a feasible solution to $L(N \setminus i, \emptyset)$. To see this, observe that

$$f(S^* \cup \widehat{S}) \leq f(S^*) + f(\widehat{S}) \leq f(S^*) + \mu_h \leq f(S^*) + \rho_i(N \setminus i) \leq f(S^* \cup i) \leq b.$$

Hence, we see that $\varphi(N \setminus i, \emptyset) - \varphi(N \setminus i, i) \geq |\widehat{S}| = h$.

2. For this part, it is sufficient to show that if the cover inequality (4) is lifted first with x_i, then $\alpha_i \leq h$. So let us consider the lifting problem $L(S, i)$. Let $S' \subseteq S$, $\left|S'\right| = |S| - 1 - h$ be such that $f(S') = \nu_{|S|-1-h}$. Then, we have

$$f(S' \cup i) = f(S') + \rho_i(S' \setminus i) \leq \nu_{|S|-1-h} + \rho_i(\emptyset) \leq b.$$

Hence, the optimal solution has at least $|S| - 1 - h$ variables positive, which shows that $\alpha_i \leq h$. □

Remark 1. Proposition 7 is a generalization of Theorem 2 of [10] given for the linear 0–1 knapsack set. Observe that for the linear case we have $\rho_i(N \setminus i) = \rho_i(\emptyset) = a_i$ for all $i \in N$ and $b - \nu_{|S|-h-1} = \mu_{h+1} - \lambda$. Thus, for a linear knapsack set, the statements of Proposition 7 reduce to: 1. if $a_i \geq \mu_h$, then $\alpha_i \geq h$; and 2. if $a_i \leq \mu_{h+1} - \lambda$, then $\alpha_i \leq h$.

Example 2. Consider the conic quadratic 0–1 knapsack set X_{CQ} given by

$$2x_1 + 1x_2 + 1.5x_3 + 0.5x_4 + a_5x_5 + \sqrt{1x_2 + 0.5x_3 + 1x_4 + c_5x_5} \leq 6.5.$$

The set $S = \{1, 2, 3, 4\}$ is a minimal cover with $\lambda = f(\{1, 2, 3, 4\}) - 6.5 = 0.08$. Therefore, $x_1 + x_2 + x_3 + x_4 \leq 3$ is valid for X. In this example we illustrate the bounds given in Proposition 7 on the lifting coefficient α_5 as a function of a_5 and c_5. Table 1 shows μ_h, ν_h, and $b - \nu_{|S|-1-h}$ for different values of h.

Given μ_h, one can solve the equation

$$\rho_5(N \setminus 5) = a_5 + \sqrt{2.5 + c_5} - \sqrt{2.5} = \mu_h,$$

to find the values for (a_5, c_5) for which the $\alpha_5 \geq h$ in all lifting sequences. In Fig. 1 we plot $a_5 + \sqrt{c_5} = \mu_h$ for $h = 1, 2, 3$ with solid lines. Similarly, solving

$$\rho_5(\emptyset) = a_5 + \sqrt{c_5} = b - \nu_{|S|-1-h},$$

one finds the values for (a_5, c_5) for which the $\alpha_5 \leq h$ in all lifting sequences. We plot the solution $h = 0, 1, 2, 3$ in Fig. 1 with dashed lines. In the same figure we show the possible values of valid lifting coefficients in different regions. Note that lifting is *sequence-independent* in areas where there is a single value; that is, the lifting coefficient for the variable is the same for any lifting order. The figure also illustrates that in a large domain of (a_5, c_5) the bounds from Proposition 7 can be used to approximate the lifting coefficient.

Please cite this article in press as: A. Atamtürk, V. Narayanan, The submodular knapsack polytope, Discrete Optimization (2009), doi:10.1016/j.disopt.2009.03.002

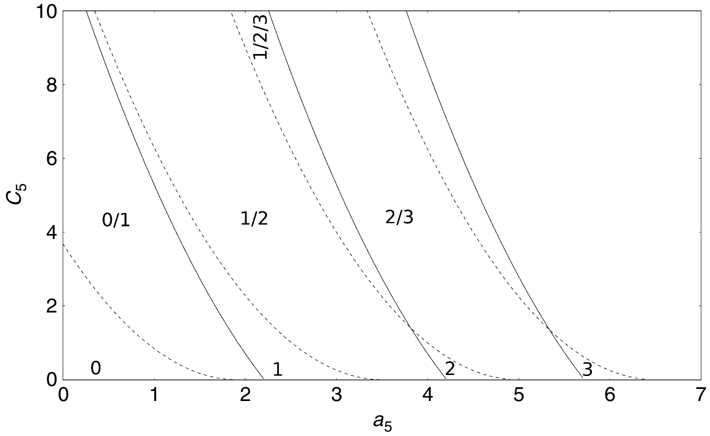

Fig. 1. Possible values for α_5 as a function of (a_5, c_5).

3. Algorithms

In this section we describe algorithms for computing the bounds in Proposition 7 on the lifting coefficients as well as for sequential lifting of the cover inequalities. We also discuss the separation problem associated with the cover inequalities. Throughout the section, we consider a submodular function of the form

$$f(S) = a(S) + g(c(S)), \tag{13}$$

where $g : \mathbb{R} \to \mathbb{R}$ is an increasing concave function and $a, c \geq \mathbf{0}$. Observe that f includes the conic quadratic function of X_{CQ} as a special case when $g(x) = \sqrt{x}$. In Section 4 we present computational results for the conic quadratic case.

3.1. Computing the bounds on the lifting coefficients

Problem (10) is the maximization of a submodular function subject to a cardinality constraint, which is $\mathcal{NP}$-hard for a general submodular function as submodular maximization includes the $\mathcal{NP}$-hard max-cut problem [31] as a special case. For a general submodular function, problem (11) is $\mathcal{NP}$-hard as it includes as a special case the min-cut problem with cardinality constraint, which is also $\mathcal{NP}$-hard [32]. For a submodular function of the form (13), we show that while the minimization problem can be solved in polynomial time, the maximization problem remains $\mathcal{NP}$-hard.

Consider, first, the cardinality-constrained minimization problem

$$v_h = \min_{T \subseteq S} \{a(T) + g(c(T)) : |T| = h\}. \tag{14}$$

Parametric linear programming is an efficient approach for minimizing a concave function over matroid constraints [33,34]. We show below that solving (14) for all $h = 1, \ldots, |S|$ can be accomplished in the same complexity as solving it for a single value of h.

Let $\mathcal{T}_h$ be the collection of subsets of S of cardinality h and

$$Y_h = \text{conv}\{(a(T), c(T)) : T \in \mathcal{T}_h\}.$$

Note that polytope $Y_h \subseteq \mathbb{R}^2_+$. Consider now the problem

$$\min \alpha + g(\gamma) : (\alpha, \gamma) \in Y_h. \tag{15}$$

Because the objective of (15) is concave, it has an optimal solution that is an extreme point of Y_h, and thus (15) is equivalent to (14). The set of candidate extreme points of Y_h can be enumerated efficiently by solving the parametric linear programming problem over Y_h

$$\min \alpha + \lambda\gamma : (\alpha, \gamma) \in Y_h \text{ for } \lambda \geq 0 \tag{16}$$

and the single optimization problem

$$\min \gamma : (\alpha, \gamma) \in Y_h. \tag{17}$$

Observe that because f is nondecreasing in on S, it is sufficient to consider extreme points that are optimal for all nonnegative objectives considered in (16) and (17). For fixed λ optimal solutions for (17) are given by the h smallest $c_i,\ i \in S$. Optimal solutions for (16) are given by the h smallest $a_i + \lambda c_i$, $i \in S$. Because the order of $(a_i + \lambda c_i),\ i \in S$, may change at most $\binom{|S|}{2}$

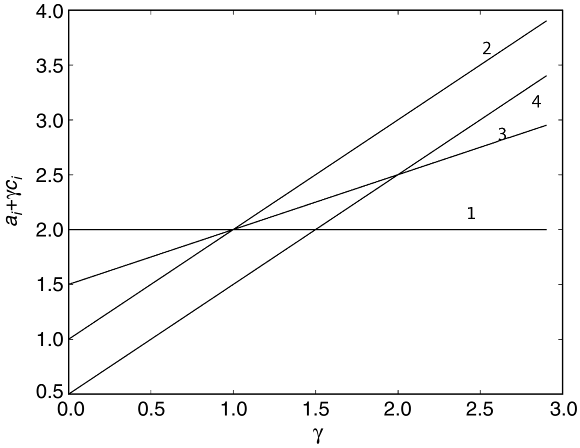

Fig. 2. Computing $\nu_h, h = 1, \ldots, 3$.

times as λ ranges over $[0, +\infty)$, there are at most $\binom{|S|}{2}$ extreme points to consider, which can be enumerated by solving (16) for each $\lambda = \lambda_{ij}$, where λ_{ij} is the solution for

$$a_i + \lambda c_i = a_j + \lambda c_j, \quad \text{for } i, j \in N : i \neq j. \tag{18}$$

As a median can be found in linear time [35], while this suggests a complexity of $O(|S|^3)$ for each ν_h, $1 \leq h \leq |S|$, a more careful analysis shows that indeed all ν_h, $h = 1, \ldots, |S|$, can be computed in the same complexity. Because the cardinality of the solutions is restricted to h, only the order changes affecting the hth smallest and $(h+1)$th smallest items are relevant for (16).

The key observation is that exchange of every pair $\{i, j\}$ is of interest for at most one value of h. Suppose, first, that there is a distinct critical value λ_{ij} for each pair. Let $\kappa_{ij} = a_i + \lambda_{ij} c_i$ and $T(\lambda_{ij}) = \left\{k \in S : a_k + \lambda_{ij} c_k < \kappa_{ij}\right\}$. Then, λ_{ij} corresponds to the two alternative optimal solutions $T(\lambda_{ij}) \cup i$ and $T(\lambda_{ij}) \cup j$ for (16) for $h = |T(\lambda_{ij})| + 1$. Suppose, now, that $p > 2$ lines intersect at λ', i.e., λ' is solution for $\binom{p}{2}$ pairs. Then, λ' corresponds to two alternative optimal solutions (an edge) for (16) for $h = |T(\lambda')| + 1, \ldots, |T(\lambda')| + p - 1$. Example 3 illustrates this observation.

Proposition 8. *There is an $O(|S|^3)$ algorithm for computing all ν_h for $h = 1, 2, \ldots, |S|$.*

Proof. There are at most $|S|$ subsets to consider as solutions for (17) for all h. On the other hand for (16), because for every pair $\{i, j\} \subseteq N$ there are at most two alternative solutions (i.e., an edge of Y_h) for a particular h, the cumulative number of extreme points (hence subsets of S) evaluated is $\binom{|S|}{2}$ for all $h = 1, 2, \ldots, |S| - 1$. As the original objective function $a(T) + g(c(T))$ can be computed in linear time for each candidate set T, the result holds. □

Example 3. Consider again the conic quadratic 0–1 knapsack set X_{CQ} in Example 2 described by the constraint

$$2x_1 + 1x_2 + 1.5x_3 + 0.5x_4 + a_5x_5 + \sqrt{1x_2 + 0.5x_3 + 1x_4 + c_5x_5} \leq 6.5.$$

Recall that $S = \{1, 2, 3, 4\}$ is a minimal cover. We apply the algorithm described above for computing ν_h, $h = 1, 2, 3$. In Fig. 2 we plot $a_i + \gamma c_i$, $i \in S$, as a function of γ. As $|S| = 4$ there are at most $\binom{4}{2} = 6$ intersections. In this example, lines 2 and 4 do not intersect.

At $\gamma = 2$ lines 3 and 4 intersect: Let ϵ be a small positive number. For $\gamma = 2 - \epsilon$, $\{1, 3\}$ and for $\gamma = 2 + \epsilon$, $\{1, 4\}$ are optimal for (16) with $h = 2$. At $\gamma = 1.5$ lines 1 and 4 intersect: For $\gamma = 1.5 - \epsilon$, $\{4\}$ and for $\gamma = 1.5 + \epsilon$, $\{1\}$ are optimal when $h = 1$. Finally, at $\gamma = 1$ lines 1, 2, and 3 intersect: For $\gamma = 1 - \epsilon$, $\{4, 2\}$ and for $\gamma = 1 + \epsilon$, $\{4, 1\}$ are optimal when $h = 2$. For $\gamma = 1 - \epsilon$, $\{4, 2, 3\}$ and for $\gamma = 1 + \epsilon$, $\{4, 1, 3\}$ are optimal when $h = 3$. Considering problem (17), we augment the list of candidate sets by $\{1, 3, 2\}$ for $h = 3$. These candidate sets and the corresponding values ν_h are listed in Table 2. The optimal set for each h is underlined in the table.

Consider now the maximization problem

$$\mu_h = \max_{T \subseteq S} \{a(T) + g(c(T)) : |T| = h\}. \tag{19}$$

Note that because the objective function is non-decreasing on S, without the cardinality constraint the problem would have the trivial solution S. We show below that, unlike in the minimization problem, with the addition of the cardinality constraint the maximization problem becomes much more difficult.

Table 2
Computing ν_h over Y_h.

h	Extreme sets	ν_h
1	{4} , {1}	1.50
2	{4, 2}, {1, 3} , {1, 4}	2.91
3	{4, 2, 3}, {4, 1, 3} , {1, 3, 2}	4.58

Proposition 9. *Problem (19) is $\mathcal{NP}$-hard.*

Proof. Ahmed and Atamtürk [8] have shown that

$$\max_{T\subseteq S} -c(T) + g(c(T)) \tag{20}$$

with $c \geq 0$ is $\mathcal{NP}$-hard (their proof can be extended to the special case where $g(x) = \sqrt{x}$ as well). We show here that (20) reduces to the cardinality restricted problem with non-decreasing objective (19): Given an instance of (20), let $\bar{c} = \max_{i\in N} c_i$ and rewrite the problem as

$$\max_{T\subseteq S} -|T|\bar{c} + \sum_{i\in T}(\bar{c} - c) + g(c(T)).$$

For all sets T because the first component of the objective is an integer multiple of $-\bar{c}$, problem (20) reduces to solving

$$-\bar{c}h + \max_{T\subseteq S}\left\{\sum_{i\in T}(\bar{c} - c_i) + g(c(T)) : |T| = h\right\}$$

for all $h = 0, 1, \ldots, |S|$ and taking the solution with the largest objective. □

Nemhauser et al. [36] give approximation algorithms for maximizing a submodular function with a cardinality constraint, which may be used to compute a lower bound on μ_h. However, in this case, we require an algorithm for computing an upper bound $\bar{\mu}_h$ so that we may employ Proposition 7; that is, if $\rho_i(N \setminus i) \geq \bar{\mu}_h$ for $\bar{\mu}_h \geq \mu_h$, then $\alpha_i \geq h$.

The algorithm we propose is similar to the one for computing ν_h. Consider the two-dimensional optimization problem

$$\bar{\mu}_h := \max \alpha + g(\gamma) : (\alpha, \gamma) \in Y_h. \tag{21}$$

Because Y_h is a relaxation of its extreme points, $\bar{\mu}_h \geq \mu_h$ holds. Note that as the objective of (19) is non-decreasing, it has an optimal solution that is a boundary point of the polytope Y_h, but not necessarily an extreme point. The objective values for the extreme points of Y_h for all $h = 1, \ldots, |S|$ can be evaluated in $O(|S|^3)$ as in the minimization case using the parametric linear programming approach.

For computing the objective for non-extreme candidate solutions of Y_h, it suffices to maximize $\alpha + g(\gamma)$ over each edge of Y_h. This problem is just a convex optimization of a univariate function: The line that goes through two adjacent extreme points (a_1, c_1) and (a_2, c_2) of Y_h is described by

$$L = \left\{(a, c) : a - a_1 = (c - c_1)\frac{a_2 - a_1}{c_2 - c_1}\right\}.$$

Substituting out a, the function

$$\bar{f}(c) = a_1 + (c - c_1)\frac{a_2 - a_1}{c_2 - c_1} + g(c),$$

is maximized at $\bar{c}$ such that $g'(\bar{c}) = \frac{a_1 - a_2}{c_2 - c_1}$. Comparing $\bar{c}$ with its bounds c_1 and c_2, we find the optimal solution over the edge. Clearly, this is accomplished in $O(1)$ for each edge of Y_h. Because the total number of adjacent extreme points considered for all h by the parametric linear programming approach is at most $\binom{|S|}{2} + |S|$, we have the following result.

Proposition 10. *There is an $O(|S|^3)$ algorithm for computing all $\bar{\mu}_h$ for $h = 1, 2, \ldots, |S|$.*

3.2. Computing the lifting coefficients

We consider now computing valid lifting coefficients for (12). Suppose the intermediate lifted inequality contains variables $x_i,\ i \in I$. Computing $\varphi(I, k),\ k \in N \setminus I$, requires solving an optimization problem over the conic quadratic 0–1 knapsack set. As this may be computationally prohibitive, we solve the continuous relaxation of the lifting problem

$$\widehat{\varphi}(I, k) = \max\left\{\sum_{i\in I}\alpha_i x_i : a_k + \sum_{i\in I} a_i x_i + g\left(c_k + \sum_{i\in I} c_i x_i\right) \leq b, \mathbf{0} \leq x \leq \mathbf{1}\right\}, \tag{22}$$

where $\alpha_i = 1$ for $i \in S \subseteq I$, to obtain the lower bound

$$\widehat{\alpha}_k = |S| - 1 - \lfloor\widehat{\varphi}(I, k)\rfloor \leq \alpha_k.$$

As α_k is integer valued, $\widehat{\alpha}_k$ is a valid lifting coefficient. In order to utilize an algorithm similar to the ones in the previous section, we restate problem (22) by exchanging the roles of the objective and constraint and let

$$\gamma_{I,k}(z) = \min\left\{a_k + \sum_{i\in I} a_i x_i + g\left(c_k + \sum_{i\in I} c_i x_i\right) : \sum_{i\in I} \alpha_i x_i \geq z,\ \mathbf{0} \leq x \leq \mathbf{1}\right\} \tag{23}$$

so that $\lfloor \widehat{\varphi}(I,k) \rfloor = \max\left\{z \in \{0,\dots,|S|-1\} : \gamma_{I,k}(z) \leq b\right\}$. Because the objective of (23) is concave, it has an optimal solution that is an extreme point of the polytope

$$R(I,z) = \left\{x \in \mathbb{R}^I : \sum_{i\in I} \alpha_i x_i \geq z,\ \mathbf{0} \leq x \leq \mathbf{1}\right\}.$$

Let $\left\{x^i : i \in K_z\right\}$ be the set of extreme points of $R(I,z)$ and

$$Z_z = \operatorname{conv}\left\{(a(x^i), c(x^i)) : i \in K_z\right\}.$$

Then, the two-dimensional optimization problem

$$\min \alpha + g(\gamma) : (\alpha,\gamma) \in Z_z$$

is equivalent to problem (23). The set of all candidate extreme points of Z_z can be enumerated by solving

$$\min \gamma : (\alpha,\gamma) \in Z_z, \tag{24}$$

and the parametric linear programming problem

$$\min \alpha + \lambda\gamma : (\alpha,\gamma) \in Z_z \quad \text{for } \lambda \geq 0, \tag{25}$$

whose optimal solution is given by a greedy algorithm that satisfies the continues knapsack constraint of $R(I,z)$ in non-decreasing order of $\frac{a_i}{\alpha_i} + \lambda\frac{c_i}{\alpha_i}$, $i \in I$. As the greedy order changes at most $\binom{|I|}{2}$ times at $\lambda = \lambda_{ij}$, where λ_{ij} solves

$$\frac{a_i}{\alpha_i} + \lambda\frac{c_i}{\alpha_i} = \frac{a_j}{\alpha_j} + \lambda\frac{c_j}{\alpha_j}, \quad \text{for } i,j \in N : i \neq j,$$

there are at most $\binom{|I|}{2}$ candidate extreme points, which are solutions to the (25) at the critical values λ_{ij}. This gives an $O(|N|^3 \log|N|)$ algorithm for each z and a total of $O(|S||N|^3 \log|N|)$ complexity for computing the lifting coefficient $\widehat{\alpha}_k$. Indeed, computing all lifting coefficients $\widehat{\alpha}_k$, $k \in N \setminus S$, can be done in the same complexity. To see this, observe that each time a new variable is introduced to the lifting problem, at most $|N|$ new critical λ_{ik} values are introduced and the continuous knapsack solutions for existing critical λ_{ij} values can be updated in $|S||N|^2 \log|N|$. Hence, we have the following result.

Proposition 11. *There is an $O(|S||N|^3 \log|N|)$ algorithm for computing all $\widehat{\alpha}_i$ for $i \in N \setminus S$.*

Even though problem (22) can be solved in strongly polynomial time as shown above, we now describe a simpler LP-based approach. Let $f_k(T) := a_k + a(T) + g(c_k + c(T))$ for $T \subseteq I$. Using the results of Atamtürk and Narayanan [25], problem (22) can be formulated as the following linear program

$$\widehat{\varphi}(I,k) = \max\left\{\sum_{i\in I} \alpha_i x_i : \pi x \leq b - f_k(\emptyset),\ \pi \in \Pi(k),\ \mathbf{0} \leq x \leq \mathbf{1}\right\}, \tag{26}$$

where $\Pi(k)$ is the set of extreme points of the extended polymatroid associated with submodular function $f_k - f_k(\emptyset)$. From polynomial equivalence of optimization and separation for polyhedra [37], linear programming problem (26) can be solved in polynomial time as the separation problem for $\pi x \leq b - f_k(\emptyset)$, $\pi \in \Pi(k)$ is an optimization problem over the extended polymatroid, which can be solved by the greedy algorithm [38].

3.3. *Solving the separation problem*

Given $x \in \mathbb{R}^N$ s.t. $\mathbf{0} \leq x \leq \mathbf{1}$, let $\bar{x} = \mathbf{1} - x$. As $\sum_{i\in C} x_i > |C| - 1$ if and only if $\sum_{i\in C} \bar{x}_i < 1$, the separation problem with respect to cover inequalities (4) can be formulated as

$$\zeta = \min\left\{\bar{x}z : az + g(cz) > b,\ z \in \{0,1\}^N\right\}. \tag{27}$$

The constraint $az + g(cz) > b$ ensures that the solution is a cover. Thus, there is a violated cover inequality if and only if $\zeta < 1$.

In order to find violated cover inequalities quickly, we employ a heuristic that rounds fractional solutions of

$$\min\left\{\bar{x}z : az + y \geq b,\ cz \geq h(y),\ \mathbf{1} \geq z \geq \mathbf{0},\ y \in \mathbb{R}\right\}, \tag{28}$$

where h is the inverse of g, to an integer solution (h exists as g is increasing). Because g is increasing concave, h is increasing convex; hence (28) is a convex optimization problem. Note that for every extreme point (y, z) of (28) there are at most two variables with $0 < z_i < 1$ and $0 < z_j < 1$. Let $-\lambda, -\mu, -\alpha, -\beta$ be the dual variables associated with the constraints, in the order listed above. Then, the first-order optimality conditions imply

$$-\bar{x}_i = -a_i\lambda - c_i\mu + \alpha_i - \beta_i, \quad i \in N,$$
$$0 = -\lambda + \mu g'(y).$$

From complementary slackness, we have

$$\bar{x}_i \begin{cases} \le a_i\lambda + c_i\mu, & \text{for } z_i = 1, \\ = a_i\lambda + c_i\mu, & \text{for } 0 < z_i < 1, \quad \text{for } i \in N. \\ \ge a_i\lambda + c_i\mu, & \text{for } z_i = 0, \end{cases} \tag{29}$$

Because there are at most two fractional z_i in extreme solutions, we compute $\binom{|N|}{2}$ candidate values for λ and μ, which are solutions for

$$\bar{x}_i = a_i\lambda + c_i\mu; \qquad \bar{x}_j = a_j\lambda + c_j\mu \quad \text{for } i, j \in N : i \neq j.$$

Assigning variables $z_i,\ i \in N$ to one in non-decreasing order of $\bar{x}_i/(a_i\lambda + c_i\mu)$ until z defines a cover for each candidate $(\lambda, \mu) \ge \mathbf{0}$, we check for the violation of the corresponding cover.

4. Computations

In this section we present our computational experiments for testing the effectiveness the inequalities for solving 0–1 programming problems with conic quadratic knapsack constraints (3). For the computational experiments we use the MIP solver of CPLEX[1] Version 11.0 that solves conic quadratic relaxations at the nodes of the branch-and-bound tree. CPLEX heuristics are turned off; all other options are kept at default values. All experiments are performed on a 3.31 GHz Pentium Linux workstation with 1 GB main memory.

In Table 3 we report the results of the experiments for varying numbers of variables (n), constraints (m), and values for Ω. For each combination, five random instances are generated with a_i from integer uniform [0, 100] and d_i from integer uniform $[0, a_i]$. The knapsack budget b is set to $0.5 \times f(N)$ So that constraints are not completely dense, we set the density of the constraints as $100 \times 2/\sqrt{n}$. The data files are available for download at http://ieor.berkeley.edu/~atamturk/data.

Even though we do not compute the bounds μ_h, ν_h on the lifting coefficients for all h because of the high computational requirement of doing so, we nevertheless, employ the easily computable special cases $\mu_1, \mu_{|S|-1}$ and $\nu_{|S|-1}$ for preprocessing before extending or lifting cover inequalities. Note that if $\rho_i(N \setminus i) \ge \mu_1$, then x_i is included in every extension. Similarly, if $\rho_i(\emptyset) \le b - \nu_{|S|-1}$, then x_i has a zero coefficient in any lifting or extension inequality; thus it is dropped from consideration. Also, if $\rho_i(N \setminus i) \ge \mu_{|S|-1}$, then α_i is fixed to $|S| - 1$ and x_i dropped from the lifting problem.

In the table we compare the integrality gap (%) of the conic quadratic relaxation, the number of cuts generated (`cuts`), the number of nodes explored (`nodes`), and the CPU time in seconds (`time`) with several cut generation options. The columns under heading CPLEX show the performance of CPLEX with no user cuts added. The other columns show the performance when cover cuts, extended cover cuts, and lifted cover cuts are added, respectively. The covers are generated only at the root node of the branch-and-bound tree using the separation algorithm explained in Section 3.3 and then extended or lifted. We employ the lifting strategy based on linear programming (26). Each row in the table is the average for five random instances. The `igap` column shows the initial integrality gap of the conic quadratic relaxation. The `rgap` columns show the integrality gap of the root relaxation after the cuts are added. CPLEX adds a small number of its own cuts, which explains the difference between the `rgap` column for CPLEX and `igap`.

We observe that as Ω increases, so does the integrality gap of the initial conic quadratic formulation. An increase in Ω weighs the nonlinear portion of the constraint more and typically leads to a higher number of fractional variables in the continuous relaxation. Note that the integrality gap increases with the number of constraints (m) and decreases with the number of variables (n); however, as expected, the number of branch-and-bound nodes and the CPU time increases with the problem size.

None of the instances with 75 variables and 20 constraints could be solved to optimality within the time limit of one hour without adding user cuts. For those instances the average remaining optimality gap at termination are 4.37%, 7.42%, and 7.76%, respectively. The addition of the cuts reduces the root gap significantly and leads to an efficient solution of all instances. As expected, extended cover cuts are more effective than just cover cuts and lifted cover cuts are more effective than extended cover cuts.

With the lifted cover cuts almost half of the instances are solved at the root node without any need for branching. On average the integrality gap is reduced from 13.9% to 0.8% for all instances. For problems that could also be solved by CPLEX (all but instances with 75 variables and 20 constraints), the average solution time is reduced from 495 s to just 7 s.

[1] CPLEX is a registered trademark of ILOG, Inc.

Please cite this article in press as: A. Atamtürk, V. Narayanan, The submodular knapsack polytope, Discrete Optimization (2009), doi:10.1016/j.disopt.2009.03.002

Table 3
Effectiveness of the submodular cover, extended cover & lifted cover cuts.

n	m	Ω	igap	CPLEX			CPLEX + covers				CPLEX + ext. covers				CPLEX + lifted covers			
				rgap	nodes	time	cuts	rgap	nodes	time	cuts	rgap	nodes	time	cuts	rgap	nodes	time
		1	14.9	10.7	209	0	16	0.1	1	0	10	0.0	0	0	15	0.0	0	0
	10	3	15.3	14.6	75	0	15	0.0	0	0	9	0.0	0	0	16	0.0	0	0
		5	16.4	16.1	98	0	16	0.0	0	0	10	0.0	0	0	17	0.0	0	0
25		1	22.6	19.8	267	1	27	0.5	3	0	19	0.0	0	0	27	0.0	0	0
	20	3	25.5	24.5	260	1	27	0.9	2	0	17	0.0	0	0	25	0.0	0	0
		5	27.5	26.9	273	1	28	0.3	1	0	17	0.0	1	0	25	0.0	0	0
		1	7.0	6.2	1,609	16	61	0.2	41	1	29	0.0	4	0	57	0.0	0	0
	10	3	9.7	9.5	2,801	21	64	0.8	94	2	35	0.1	7	1	59	0.0	0	0
		5	10.7	10.7	3,239	24	68	0.8	51	1	34	0.0	2	1	76	0.0	1	1
50		1	12.5	11.1	47,316	860	111	1.9	476	18	49	1.3	393	16	179	1.4	133	8
	20	3	17.6	17.5	41,290	623	128	1.7	244	11	67	0.4	23	2	191	0.2	20	4
		5	21.8	21.8	45,009	606	139	1.0	45	4	62	0.5	19	2	180	0.6	39	5
		1	4.1	4.1	85,153	1600	121	1.6	1,600	52	50	0.8	865	30	161	1.1	1127	40
	10	3	6.0	6.0	91,619	1617	162	2.1	3,792	131	59	1.3	1,595	59	227	0.9	278	18
		5	7.5	7.5	111,634	2064	185	2.5	4,240	152	67	1.7	1,280	52	253	1.3	586	33
75		1	7.2	7.1	77,164	3648[a]	144	3.8	25,229	2102	65	2.8	11,273	928	440	2.5	3664	523
	20	3	11.0	11.0	86,691	3651[a]	228	6.3	28,524	2228	82	4.0	14,860	1252	478	3.1	3466	634
		5	13.5	13.5	87,150	3650[a]	286	6.2	13,223	1223	104	3.5	4,747	480	486	2.8	2507	478

[a] Instances not solved to optimality within an hour time limit.

Over all instances, the average number of nodes is 4309, 1948, and 657 with cover cuts, extended covers cuts, and lifted cuts, respectively. On the other hand, the average CPU time (in seconds) is 496, 157, and 97 with cover cuts, extended covers cuts, and lifted cuts, respectively. As is typical in numerical studies with cuts, we see a decreasing marginal rate of improvement with the biggest marginal impact achieved for the simpler cuts.

Acknowledgement

This research has been supported, in part, by Grant # DMI0700203 from the National Science Foundation.

References

[1] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley and Sons, New York, 1988.
[2] A. Charnes, W.W. Cooper, Chance-constrained programming, Management Science 6 (1959) 73–79.
[3] J.R. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer, Berlin, 1997.
[4] D. Bertsimas, I. Popescu, Optimal inequalities in probability theory: A convex optimization approach, SIAM Journal on Optimization 15 (2005) 78–804.
[5] L. El Ghaoui, M. Oks, F. Oustry, Worst-case value-at-risk and robust portfolio optimization: A conic programming approach, Operations Research 51 (2003) 543–556.
[6] A. Ben-Tal, A. Nemirovski, Robust optimization–methodology and applications, Mathematical Programming 92 (2002) 453–480.
[7] A. Ben-Tal, A. Nemirovski, Robust solutions of uncertain linear programs, Operations Research Letters 25 (1999) 1–13.
[8] S. Ahmed, A. Atamtürk, Maximizing a class of submodular utility functions, Research Report BCOL.08.02, IEOR, University of California-Berkeley, March 2008.
[9] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, John Wiley & Sons, Inc., New York, 1990.
[10] E. Balas, Facets of the knapsack polytope, Mathematical Programming 8 (1975) 146–164.
[11] P.L. Hammer, E.L. Johnson, U.N. Peled, Facets of regular 0–1 polytopes, Mathematical Programming 8 (1975) 179–206.
[12] L.A. Wolsey, Faces for linear inequality in 0–1 variables, Mathematical Programming 8 (1975) 165–178.
[13] M.W. Padberg, $(1, k)$-configurations and facets for packing problems, Mathematical Programming 18 (1980) 94–99.
[14] Z. Gu, G.L. Nemhauser, M.W.P. Savelsbergh, Cover inequalities for 0–1 integer programs: Computation, INFORMS Journal on Computing 10 (1998) 427–437.
[15] A. Atamtürk, On the facets of mixed–integer knapsack polyhedron, Mathematical Programming 98 (2003) 145–175.
[16] A. Atamtürk, Cover and pack inequalities for (mixed) integer programming, Annals of Operations Research 139 (2005) 21–38.
[17] D. Hochbaum, A nonlinear knapsack problem, Operations Research Letters 17 (1995) 103–110.
[18] A. Melman, G. Rabinowitz, An efficient method for a class of continuous nonlinear knapsack problems, SIAM Review 42 (2000) 440–448.
[19] T.L. Morin, R.E. Marsten, An algorithm for nonlinear knapsack problems, Management Science 22 (1976) 1147–1158.
[20] A. Billionnet, F. Calmels, Linear programming for the 0–1 quadratic knapsack problem, European Journal of Operational Research 92 (1996) 310–325.
[21] A. Caprara, D. Pisinger, P. Toth, Exact solution of the quadratic knapsack problem, INFORMS Journal on Computing 11 (1999) 125–137.
[22] C. Helmberg, F. Rendl, R. Weismantel, A semidefinite programming approach to the quadratic knapsack problem, Journal of Combinatorial Optimization 4 (2000) 197–215.
[23] G. Gallo, B. Simeone, On the supermodular knapsack problem, Mathematical Programming 45 (1989) 295–309.
[24] M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, Operations Research Letters 32 (2004) 41–43.
[25] A. Atamtürk, V. Narayanan, Polymatroids and risk minimization in discrete optimization, Operations Research Letters 36 (2008) 618–622.
[26] K.M. Bretthauer, B. Shetty, The nonlinear knapsack problem—algorithms and applications, European Journal of Operational Research 138 (2002) 459–472.
[27] A. Atamtürk, V. Narayanan, Conic mixed-integer rounding cuts. Research Report BCOL.06.03, IEOR, University of California-Berkeley, December 2006 Mathematical Programming, forthcoming (doi:10.1007/s10107-008-0239-4).
[28] M.T. Çezik, G. Iyengar, Cuts for mixed 0–1 conic programming, Mathematical Programming 104 (2005) 179–202.
[29] E. Balas, E. Zemel, Facets of the knapsack polytope from minimal covers, SIAM Journal of Applied Mathematics 34 (1978) 119–148.

Please cite this article in press as: A. Atamtürk, V. Narayanan, The submodular knapsack polytope, Discrete Optimization (2009), doi:10.1016/j.disopt.2009.03.002

[30] E. Balas, E. Zemel, Lifting and complementing yields all facets of positive zero–one programming polytopes, in: R.W. Cottle et al., (Eds.), Proceedings of the International Conference on Mathematical Programming, pp. 13–24, 1984.
[31] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, 1972, pp. 85–103.
[32] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete graph problems, Theoretical Computer Science 1 (1976) 237–267.
[33] R. Hassin, A. Tamir, Maximizing classes of two-parameter objectives over matroids, Mathematics of Operations Research 14 (1989) 362–375.
[34] H. Ishii, S. Shiode, T. Nishida, Stochastic spanning tree problem, Discrete Applied Mathematics 3 (1981) 263–273.
[35] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd ed., MIT Press, Cambridge, MA, 2001.
[36] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions-I, Mathematical Programming 14 (1978) 265–294.
[37] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (1981) 169–197.
[38] J. Edmonds, Submodular functions, matroids and certain polyhedra, in: R. Guy (Ed.), in: Combinatorial Structures and their Applications, vol. 11, Gordon and Breach, New York, NY, 1971, pp. 69–87.

Please cite this article in press as: A. Atamtürk, V. Narayanan, The submodular knapsack polytope, Discrete Optimization (2009), doi:10.1016/j.disopt.2009.03.002