A Polyhedral Study of the Cardinality Constrained Knapsack Problem

I.R. de Farias, Jr.*
CORE, 34 Voie du Roman Pays
1348 Louvain-la-Neuve, Belgium
defarias@core.ucl.ac.be

G.L. Nemhauser*
School of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0205
george.nemhauser@isye.gatech.edu

December 26, 2001

Abstract

A cardinality constrained knapsack problem is a continuous knapsack problem in which no more than a specified number of nonnegative variables are allowed to be positive. This structure occurs, for example, in areas such as finance, location, and scheduling. Traditionally, cardinality constraints are modeled by introducing auxiliary 0-1 variables and additional constraints that relate the continuous and the 0-1 variables. We use an alternative approach, in which we keep in the model only the continuous variables, and we enforce the cardinality constraint through a specialized branching scheme and the use of strong inequalities valid for the convex hull of the feasible set in the space of the continuous variables. To derive the valid inequalities, we extend the concepts of cover and cover inequality, commonly used in 0-1 programming, to this class of problems, and we show how cover inequalities can be lifted to derive facet-defining inequalities. We present three families of non-trivial facet-defining inequalities that are lifted cover inequalities. Finally, we report computational results that demonstrate the effectiveness of lifted cover inequalities and the superiority of the approach of not introducing auxiliary 0-1 variables over the traditional MIP approach for this class of problems.

Keywords: mixed-integer programming, knapsack problem, cardinality constrained programming, branch-and-cut

*Partially supported by NSF grants DMI-0100020 and DMI-0121495.

1 Introduction

Let n and K be two positive integers, and $N = \{1, \ldots, n\}$. For each $j \in N$, let u_j be a positive number. The *cardinality constrained knapsack problem* (CCKP) is

$$\begin{aligned} \max \quad & \textstyle\sum_{j \in N} c_j x_j & \\ & \textstyle\sum_{j \in N} a_j x_j \le b & (1) \\ & \text{at most } K \text{ variables can be positive} & (2) \\ & x_j \le u_j, \quad j \in N & (3) \\ & x_j \ge 0, \quad j \in N. & (4) \end{aligned}$$

We denote by S the set of feasible solutions of CCKP, i.e. $S = \{x \in \Re^n : x$ satisfies (1)-(4)$\}$, $PS = \text{conv}(S)$, and LPS is the set of feasible solutions of the LP relaxation, i.e. $LPS = \{x \in \Re^n : x$ satisfies (1), (3), and (4)$\}$. We assume that:

1. $a_1 \ge \cdots \ge a_n$
2. $b > 0$ and $a_j \ge 0 \ \forall j \in N$
3. a_j is scaled such that $u_j = 1$ and $a_j \le b \ \forall j \in N$
4. $\sum_{j=1}^{K} a_j > b$
5. $2 \le K \le n - 1$.

We can assume 1. without loss of generality. Once 2. is assumed, 3. can be assumed without loss of generality. When 4. does not hold, (1) is redundant. The case $K = 1$ is discussed in [13], and when $K \ge n$, (2) is redundant.

We now establish the complexity of CCKP.

Theorem 1 *CCKP is NP-hard.*

Proof We reduce the partition problem [15] to CCKP.
<u>**PARTITION**</u>
INSTANCE: Two sets of positive integers $N = \{1, \ldots, n\}$ and $S = \{a_1, \ldots, a_n\}$.
QUESTION: Does N have a subset N' such that $\sum_{i \in N'} a_i = \frac{1}{2} \sum_{i \in N} a_i$?

The solution to PARTITION is "yes" iff the optimal value to CCKP(K):

$$\begin{aligned} \max \quad & \textstyle\sum_{j \in N} a_j x_j + \sum_{j \in N} x_j \\ & \textstyle\sum_{j \in N} a_j x_j \le \frac{1}{2} \sum_{j \in N} a_j \\ & \text{at most } K \text{ variables can be positive} \\ & 0 \le x_j \le 1, \quad j \in N \end{aligned}$$

is $\frac{1}{2}\sum_{j\in N} a_j + K$ for some $K \in \{1,\ldots,n\}$. Thus, by solving CCKP(1),...,CCKP(n), we can solve PARTITION. □

Constraint (2) is present in a large number of applications, such as portfolio optimization [6, 27], p-median [10], synthesis of process networks [5, 28], etc. It is usually modeled by introducing 0-1 variables $y_j, j \in N$, and the constraints

$$x_j \le u_j y_j, j \in N, \tag{5}$$
$$\textstyle\sum_{j\in N} y_j \le K, \tag{6}$$

see [8, 22].

Rather than introducing auxiliary 0-1 variables and the inequalities (5) and (6) to model (2), we keep only the continuous variables, and we enforce (6) algorithmically, directly in the branch-and-cut algorithm by using a specialized branching scheme and strong inequalities valid for *PS* (which is defined in the space of the continuous variables). The idea of dispensing with auxiliary 0-1 variables to model certain combinatorial constraints on continuous variables and enforcing the combinatorial constraints directly in the branch-and-bound algorithm through a specialized branching scheme, was pioneered by Beale and Tomlin [3, 4] in the context of special ordered sets (SOS) of types I and II.

For several NP-hard combinatorial optimization problems, branch-and-cut has proven to be more effective than a branch-and-bound algorithm that does not account for the polyhedral structure of the convex hull of the set of feasible solutions of the problem, see [7, 16, 17, 19, 21, 24, 26, 30, 33]. In this paper we study the facetial structure of *PS*, with the purpose of using strong inequalities valid for *PS* as cuts in a branch-and-cut scheme without auxiliary 0-1 variables for the *cardinality constrained optimization problem* (CCOP):

$$\begin{array}{ll} \max & \sum_{j\in N} c_j x_j \\ & \sum_{j\in N} a_{ij} x_j \le b_i, \quad i \in M \\ & x \text{ satisfies (2)-(4),} \end{array} \tag{7}$$

where $M = \{1,\ldots,m\}$ and m is a positive integer.

Some potential benefits of this approach are [12]:

- Faster LP relaxations. Adding the auxiliary 0-1 variables and the new constraints substantially increases the size of the model. Also, the inclusion of variable upper bound constraints, such as (5), may turn the LP relaxation into a highly degenerate problem.

- Less enumeration. It is easy to show that a relaxation of the 0-1 mixed integer problem may have fractional basic solutions that satisfy (2). In this case, even though the solution satisfies the cardinality constraint, additional branching may be required.

This approach has been studied by de Farias [9], and recently it has been used by Bienstock in the context of portfolio optimization [6], and by de Farias, Johnson, and Nemhauser in the context of complementarity problems [13], and the generalized assignment problem [11, 14]. It has also been explored in the context of logical programming, see for example [12, 20].

Let N_0 and N_1 be two disjoint subsets of N and $P \subseteq \Re^n$. Throughout the paper we denote $P(N_0, N_1) = P \cap \{x \in \Re^n : x_j = 0 \ \forall j \in N_0 \text{ and } x_j = 1 \ \forall j \in N_1\}$. For a polyhedron P we denote by $V(P)$ the set of vertices of P. We make use of the following well-known result. We drop Assumption 1. here.

Proposition 1 *Consider the continuous knapsack problem*

$$max\{\sum_{j\in N} c_j x_j : \sum_{j\in N} a_j x_j \leq b \text{ and } x \in [0,1]^n\}. \tag{8}$$

Suppose that $\frac{c_1}{a_1} \geq \cdots \geq \frac{c_n}{a_n}$, $\sum_{j=1}^{l-1} a_j < b$, *and* $\sum_{j=1}^{l} a_j \geq b$, *for some* $l \in N$. *Then,* x^* *given by*

$$x_j^* = \begin{cases} 1 & \text{if } j = 1, \ldots, l-1 \\ \frac{b - \sum_{j=1}^{l-1} a_j}{a_l} & \text{if } j = l \\ 0 & \text{otherwise.} \end{cases}$$

is an optimal solution to (8). □

The organization of the paper is as follows. In Section 2 we present the trivial facet-defining inequalities for *PS*. We give a necessary and sufficient condition for the trivial facet-defining inequalities to completely characterize *PS*. In Section 3 we extend the concepts of cover and cover inequality, commonly used in 0-1 programming [1, 18, 31], to derive facet-defining inequalities for lower-dimensional projections of *PS*. We present three nontrivial families of facet-defining inequalities for *PS* that can be obtained by lifting these cover inequalities. In Section 4 we report computational results that demonstrate the effectiveness of lifted cover inequalities and the superiority of the approach of not introducing auxiliary 0-1 variables over the traditional MIP approach to solve difficult instances of CCOP. Finally, in Section 5 we discuss directions for further research.

2 Trivial Facet-Defining Inequalities

In this section we discuss some families of facet-defining inequalities for *PS* that are easily implied by the problem, and which we call trivial. We also present a necessary and sufficient condition for them to completely characterize *PS*.

The proofs of Propositions 2-4 are easy, and are omitted.

Proposition 2 *PS is full-dimensional.* □

Proposition 3 *Let $N_0, N_1 \subseteq N$ with $N_0 \cap N_1 = \emptyset$, and suppose that $S(N_0, N_1) \neq \emptyset$. Any vertex of LPS$(N_0, N_1)$ has at most one fractional component. Moreover, a point with a fractional component can be a vertex of LPS(N_0, N_1) only if it satisfies (1) at equality. Also, the vertices of PS(N_0, N_1) are the vertices of LPS(N_0, N_1) that satisfy (2).* □

Proposition 4 *Inequality (1) is facet-defining iff $\sum_{j=1}^{K-1} a_j + a_n \geq b$. Inequality (3) is facet-defining iff $a_j < b, j \in N$. Inequality (4) is facet-defining $\forall j \in N$.* □

Example 1 Let $n = 4$, $K = 2$, and (1) be

$$6x_1 + 4x_2 + 3x_3 + x_4 \leq 6. \tag{9}$$

Then, $x_j \leq 1\ \forall j \in \{2, 3, 4\}$, $x_j \geq 0\ \forall j \in \{1, 2, 3, 4\}$, and (9) are facet-defining. Note that (9) is stronger than $x_1 \leq 1$, which, therefore, is not facet-defining. □

We now give a necessary and sufficient condition for *PS* = *LPS*.

Proposition 5 *PS = LPS iff $\sum_{j=n-K+1}^{n} a_j \geq b$.*

Proof If $\sum_{j=n-K+1}^{n} a_j < b$, then $\hat{x}$ given by

$$\hat{x}_j = \begin{cases} 1 & \text{if } n - K + 1 \leq j \leq n \\ \min\{1, \frac{b - \sum_{s=n-K+1}^{n} a_s}{a_j}\} & \text{if } j = n - K \\ 0 & \text{otherwise} \end{cases}$$

is a vertex of *LPS* that does not satisfy (2).

From Proposition 3, we know that any vertex of *LPS* that does not satisfy (2) must have at least K components equal to 1. Because $a_{n-K+1}, \ldots, a_n$ are the K smallest knapsack coefficients, this implies that *LPS* has a vertex that does not satisfy (2) only if $\sum_{j=n-K+1}^{n} a_j < b$. □

When *PS* ≠ *LPS*, all vertices of *LPS* that do not satisfy (2) can be cut off by a single inequality. This inequality is presented in the next proposition.

Proposition 6 *The inequality*

$$\sum_{j \in N} x_j \leq K \tag{10}$$

is facet-defining iff $a_1 + \sum_{j=n-K+2}^{n} a_j \leq b$ and $\sum_{j=n-K}^{n-1} a_j \leq b$.

Proof If $a_1 + \sum_{j=n-K+2}^{n} a_j > b$, $x_1 = 0\ \forall x \in S$ such that $\sum_{j\in N} x_j = K$. This means that (10) defines a facet only if $a_1 + \sum_{j=n-K+2}^{n} a_j \leq b$. If $\sum_{j=n-K}^{n-1} a_j > b$, $x_n = 1\ \forall x \in S$ such that $\sum_{j\in N} x_j = K$. Thus, (10) defines a facet only if $\sum_{j=n-K}^{n-1} a_j \leq b$.

It is easy to show that when $a_1 + \sum_{j=n-K+2}^{n} a_j \leq b$, and $\sum_{j=n-K}^{n-1} a_j \leq b$, S has n linearly independent points that satisfy (10) at equality. □

Example 2 Let $n = 5$, $K = 2$, and (1) be

$$4x_1 + 3x_2 + 2x_3 + x_4 + x_5 \leq 6. \tag{11}$$

Then, $\sum_{j=1}^{5} x_j \leq 2$ is facet-defining. On the other hand, (10) does not define a facet for *PS* in Example 1 □

Proposition 7 *Inequality (10) cuts off all vertices of LPS that do not satisfy (2).*

Proof Let $\hat{x}$ be a vertex of *LPS* that does not satisfy (2), i.e. $\hat{x}$ has more than K positive components. From Proposition 3 we know that at least K of these positive components are equal to 1. Therefore, $\sum_{j\in N} \hat{x}_j > K$. □

We now give a necessary and sufficient condition for the system defined by (1), (3), (4), and (10) to define *PS*.

Proposition 8 *PS* $= LPS \cap \{x : x$ *satisfies* $(10)\}$ *iff* $a_1 + \sum_{j\in T-\{t\}} a_j \leq b\ \forall T \subseteq N - \{1\}$ *such that* $|T| = K$, $\sum_{j\in T} a_j < b$, *and* $a_t = min\{a_j : j \in T\}$ *(in case there are two different indices* t *for which* $a_t = min\{a_j : j \in T\}$*, choose one arbitrarily).*

Proof Let $T \subseteq N - \{1\}$ be such that $|T| = K$, $\sum_{j\in T} a_j < b$, and let $a_t = \min\{a_j : j \in T\}$. By Assumption 4. of Section 1, $\sum_{j=1}^{K} a_j > b$. Thus, $a_1 > a_t$. Suppose that $a_1 + \sum_{j\in T-\{t\}} a_j > b$. Then, $\hat{x}$ given by

$$\hat{x}_j = \begin{cases} 1 & \text{if } j \in T - \{t\} \\ \frac{b - \sum_{r\in T} a_r}{a_1 - a_t} & \text{if } j = 1 \\ \frac{a_1 + \sum_{r\in T-\{t\}} a_r - b}{a_1 - a_t} & \text{if } j = t \\ 0 & \text{otherwise} \end{cases}$$

is a vertex of $LPS \cap \{x : x \text{ satisfies } (10)\}$ that does not satisfy (2).

Suppose that $LPS \cap \{x : x \text{ satisfies } (10)\}$ has a vertex $\tilde{x}$ that does not satisfy (2). Because there are no other inequalities in the description of $LPS \cap \{x : x \text{ satisfies } (10)\}$ besides (1), (3), (4), and (10), $\tilde{x}$ can have at most two fractional components. Therefore, it must have at least $K-1$ components equal to 1. Because of (10), $\tilde{x}$ cannot have more than $K-1$ components equal to 1. This means that $\tilde{x}$ has two fractional components and $K-1$ components equal to 1. Let the fractional components be $\tilde{x}_u$ and $\tilde{x}_v$, and let U be the set of indices of all components of $\tilde{x}$ that are equal to 1. Then,

$$\tilde{x}_u + \tilde{x}_v = 1 \tag{12}$$

and

$$a_u\tilde{x}_u + a_v\tilde{x}_v = b - \sum_{j\in U} a_j. \tag{13}$$

The system defined by (12) and (13) has a unique solution only if $a_u \neq a_v$. Suppose, without loss of generality, that $a_u < a_v$. Then, clearly $b - \sum_{j\in U} a_j - a_u > 0$. If $1 \notin U$, define $T = U \cup \{u\}$. If $1 \in U$, define $T = (U - \{1\}) \cup \{u, v\}$. In any case, $\sum_{j\in T} a_j < b$, $|T| = K$, and $1 \notin T$. However, because $b - \sum_{j\in U} a_j - a_v < 0$ and $a_v \leq a_1$, if $a_t = \min\{a_j : j \in T\}$, $a_1 + \sum_{j\in T-\{t\}} a_j > b$. □

3 Sequentially Lifted Cover Inequalities

In this section we study facet-defining inequalities for *PS* that are derived by applying the sequential lifting procedure [25, 32] to a family of inequalities that define facets for $PS(N_0, N_1)$ for some $N_0, N_1 \subset N$ with $N_0 \cap N_1 = \emptyset$. We call these inequalities *cover inequalities*. They are defined by sets of indices that we call *covers*. Our definition of a cover is based on the similar concept used in 0-1 programming [1, 18, 31]. A major difference is that in our case the cover inequalities are valid for *LPS*, whereas in 0-1 programming they are not valid for the feasible set of the LP relaxation. However, by lifting our cover inequalities, we obtain facet-defining inequalities for *PS* that are not valid for *LPS*, and therefore can be used as cuts in a branch-and-cut scheme to solve CCOP. We present three families of facet-defining inequalities for *PS* obtained by lifting cover inequalities in a specific order.

Definition *Let C, N_0, and N_1 be three disjoint subsets of N with $N = C \cup N_0 \cup N_1$ and $|C| = K - |N_1|$. If $\sum_{j\in C} a_j > b - \sum_{j\in N_1} a_j$, we say that C is a cover for $PS(N_0, N_1)$, and that*

$$\sum_{j\in C} a_j x_j \leq b - \sum_{j\in N_1} a_j \tag{14}$$

is a cover inequality for $PS(N_0, N_1)$. □

It is easy to show that

Proposition 9 *Inequality (14) is valid and facet-defining for $PS(N_0, N_1)$.* □

Example 1 (Continued) The set $\{1\}$ is a cover for $PS(\{2,3\},\{4\})$, and the cover inequality is

$$6x_1 \leq 5. \tag{15}$$

□

Inequality (14) can be sequentially lifted to generate facet-defining inequalities for PS. The lifting procedure is based on Lemma 1, which is adapted from [32], Theorem 1; see [9] for a proof.

Lemma 1 *Let $P \subset \Re^d$ be a polytope. Define $l_j = min\{z_j : z \in P\}$ and $u_j = max\{z_j : z \in P\}$, $j = 1, \ldots, d$. Let $\tilde{z} \in P$, and suppose that*

$$\sum_{j=1}^{d-1} \alpha_j z_j \leq \beta, \tag{16}$$

is a valid inequality for $P \bigcap \{z \in \Re^d : z_d = \tilde{z}_d\}$. Define

$$\alpha_d^{max} = \begin{cases} min\{\frac{\sum_{j=1}^{d-1} \alpha_j z_j - \beta}{\tilde{z}_d - z_d} : z \in V(P) \text{ and } z_d > \tilde{z}_d\} & \text{if } \tilde{z}_d < u_d \\ \infty & \text{if } \tilde{z}_d = u_d \end{cases}$$

and

$$\alpha_d^{min} = \begin{cases} max\{\frac{\sum_{j=1}^{d-1} \alpha_j z_j - \beta}{\tilde{z}_d - z_d} : z \in V(P) \text{ and } z_d < \tilde{z}_d\} & \text{if } \tilde{z}_d > l_d \\ -\infty & \text{if } \tilde{z}_d = l_d. \end{cases}$$

Then

$$\sum_{j=1}^{d-1} \alpha_j z_j + \alpha_d z_d \leq \beta + \alpha_d \tilde{z}_d \tag{17}$$

is a valid inequality for P if and only if $\alpha_d^{min} \leq \alpha_d \leq \alpha_d^{max}$. Moreover, if (16) defines a face of $P \bigcap \{z \in \Re^d : z_d = \tilde{z}_d\}$ of dimension t, and $\alpha_d = \alpha_d^{min} > -\infty$, or $\alpha_d = \alpha_d^{max} < \infty$, then (17) defines a face of P of dimension at least $t+1$. □

In our case, when $\tilde{z}_d = 1$, $\alpha_d^{\max} = \infty$, and the lifting coefficient is given by $\alpha_d^{\min}$. When $\tilde{z}_d = 0$, $\alpha_d^{\min} = -\infty$, and the lifting coefficient is given by $\alpha_d^{\max}$. As lifting is always possible when $\tilde{z}_d \in \{0,1\}$, we will fix variables at 0 and 1 only. We leave it as an open question whether it is possible to derive additional facet-defining inequalities by fixing variables at fractional values.

It can be easily shown that

Proposition 10 *Suppose we lift the cover inequality (14) with respect to $x_j, j \in N_0 \cup N_1$, in a certain order. Let $r, s \in N_0$, and α_r and α_s be the lifting coefficients of x_r and x_s, respectively, where the inequality has been lifted with respect to x_r immediately before x_s. Suppose we exchange the lifting order of x_r and x_s but we keep the lifting order of the variables lifted before x_r and x_s. If α'_r and α'_s are the new lifting coefficients of x_r and x_s, respectively, then $\alpha_r \geq \alpha'_r$ and $\alpha_s \leq \alpha'_s$.*

Likewise, if $u, v \in N_1$, α_u and α_v are the lifting coefficients of x_u and x_v, respectively, and the inequality has been lifted with respect to x_u immediately before x_v, then the new lifting coefficients α'_u and α'_v of x_u and x_v, respectively, obtained by exchanging the lifting order of x_u and x_v, but keeping the lifting order of the variables lifted earlier, are such that $\alpha_u \leq \alpha'_u$ and $\alpha_v \geq \alpha'_v$. □

Given a cover inequality that is facet-defining for $PS(N_0, N_1)$ with $N_0 \neq \emptyset$, we show next how the inequality can be lifted with respect to a variable x_l, $l \in N_0$, to give a facet-defining inequality for $PS(N_0 - \{l\}, N_1)$ that is not valid for $LPS(N_0 - \{l\}, N_1)$.

Proposition 11 *Let C, N_0, and N_1 be three disjoint subsets of N with $N = C \cup N_0 \cup N_1$ and $|C| = K - |N_1|$. Suppose $N_0 \neq \emptyset$ and C is a cover for $PS(N_0, N_1)$. Let $i \in C$ and $l \in N_0$ be such that*

$$a_i = min\{a_j : j \in C\} \tag{18}$$

(in case there are several coefficients satisfying (18), choose one arbitrarily) and

$$\sum_{j \in C - \{i\}} a_j + a_l < b - \sum_{j \in N_1} a_j. \tag{19}$$

Then,

$$\sum_{j \in C} a_j x_j + (b - \sum_{j \in N_1} a_j - \sum_{j \in C - \{i\}} a_j) x_l \leq b - \sum_{j \in N_1} a_j \tag{20}$$

defines a facet of $PS(N_0 - \{l\}, N_1)$.

Proof We prove the proposition by lifting (14) with respect to x_l. Let α_l be the lifting coefficient, i.e.

$$\alpha_l = \min\{\frac{b - \sum_{j \in N_1} a_j - \sum_{j \in C} a_j x_j}{x_l} : x \in V(PS(N_0 - \{l\}, N_1)) \text{ and } x_l > 0\}.$$

Let $\hat{x} \in V(PS(N_0 - \{l\}, N_1))$ with $\hat{x}_l > 0$. Since at most K variables can be positive and $\hat{x}_l > 0$, at most $K - 1$ components $\hat{x}_j, j \in C$, of $\hat{x}$ can be positive. Thus,

$$\sum_{j \in C} a_j \hat{x} \leq \sum_{j \in C - \{i\}} a_j.$$

Because of (19), $\hat{x}$ must then satisfy (1) strictly at inequality. Thus, from Proposition 3, $\hat{x}_l$ cannot be fractional, and therefore must be equal to 1. This means that

$$\alpha_l = (b - \sum_{j \in N_1} a_j) - \max\{\sum_{j \in C} a_j x_j : x \in V(PS(N_0 - \{l\}, N_1)) \text{ and } x_l = 1\} = b - \sum_{j \in N_1} a_j - \sum_{j \in C - \{i\}} a_j.$$

□

Example 1 (Continued) Lifting (15) with respect to x_2, we obtain

$$6x_1 + 5x_2 \leq 5, \tag{21}$$

which defines a facet of $PS(\{3\}, \{4\})$. Now we apply Lemma 1 to complete the lifting. Lifting (21) with respect to x_4, the lifting coefficient is

$$\alpha_4 = \max\{\frac{6x_1 + 5x_2 - 5}{1 - x_4} : x \in V(PS(\{3\}, \emptyset)) \text{ and } x_4 < 1\}.$$

If $\hat{x} \in V(PS(\{3\}, \emptyset))$ is such that $\hat{x}_4 < 1$ and

$$\alpha_4 = \frac{6\hat{x}_1 + 5\hat{x}_2 - 5}{1 - \hat{x}_4},$$

then clearly $\hat{x}_1 > 0$. Because of Proposition 3, $\hat{x}_1 = 1$ if $\hat{x}_4$ is fractional. However, from (9), $\hat{x}_1 = 1 \Rightarrow \hat{x}_4 = 0$. Therefore, $\hat{x}_4 = 0$. So,

$$\alpha_4 = 6\hat{x}_1 + 5\hat{x}_2 - 5.$$

From Proposition 1, $\hat{x}_1 = \frac{1}{3}$ and $\hat{x}_2 = 1$. Therefore, $\alpha_4 = 2$, and

$$6x_1 + 5x_2 + 2x_4 \leq 7 \tag{22}$$

defines a facet of $PS(\{3\}, \emptyset)$. The lifting coefficient of x_3 is

$$\alpha_3 = \min\{\frac{7 - 6x_1 - 5x_2 - 2x_4}{x_3} : x \in V(PS) \text{ and } x_3 > 0\}.$$

If $\tilde{x} \in V(PS)$ is such that $\tilde{x}_3 > 0$ and

$$\alpha_3 = \frac{7 - 6\tilde{x}_1 - 5\tilde{x}_2 - 2\tilde{x}_4}{\tilde{x}_3},$$

then at most one of $\tilde{x}_1$, $\tilde{x}_2$, and $\tilde{x}_4$ can be positive. If $\tilde{x}_3 = 1$, either $\tilde{x}_4 = 1$ and $\alpha_3 = 5$, or, because of Proposition 1, $\tilde{x}_2 = \frac{3}{4}$, and $\alpha_3 = 3.25$. If $\tilde{x}_3$ is fractional, then $\tilde{x}_2 = 1$, and $\alpha_3 = 3$. Therefore, $\alpha_3 = 3$, and

$$6x_1 + 5x_2 + 3x_3 + 2x_4 \leq 7 \tag{23}$$

defines a facet of PS. □

Note that Proposition 11 still holds when $\sum_{j\in C-\{i\}} a_j + a_l = b - \sum_{j\in N_1} a_j$. But then, $\alpha_l = a_l$. In this case, one could have started with $PS(N_0-\{l\}, N_1)$ as the projected polytope, and $C\cup\{l\}$ as the cover. Also, if $\sum_{j\in C-\{i\}} a_j + a_l > b - \sum_{j\in N_1} a_j$, and (14) is lifted first with respect to x_l, the lifting coefficient is a_l. In the same way, if (14) is lifted first with respect to a variable x_k with $k \in N_1$, the lifting coefficient is a_k. Therefore, when studying sequentially lifted cover inequalities for PS, it suffices to start with (20). Inequality (20) resembles the *fundamental complementarity inequalities* defined in [13].

An immediate consequence of Proposition 11 is that

Corollary 1 *Let $C \subset N$ be such that $|C| = K$ and $\sum_{j\in C} a_j > b$. Let $i \in C$ and $l \in N - C$ be such that $a_i = min\{a_j : j \in C\}$ and $\sum_{j\in C-\{i\}} a_j + a_l < b$. Then,*

$$\sum_{j\in C} a_j x_j + (b - \sum_{j\in C-\{i\}} a_j) x_l \le b \tag{24}$$

is valid for PS. □

Example 1 (Continued) The inequality

$$4x_2 + 3x_3 + 2x_4 \le 6 \tag{25}$$

is valid for PS by Corollary 1. Note that (25) cuts off the point $\hat{x}$ given by $\hat{x}_1 = 0$, $\hat{x}_2 = \frac{1}{2}$, $\hat{x}_3 = \hat{x}_4 = 1$, which is a vertex of LPS, and therefore, (25) is not valid for LPS.

Inequality (25) defines a facet of $PS(\{1\}, \emptyset)$. Lifting (25) with respect to x_1, we obtain

$$\frac{24}{5}x_1 + 4x_2 + 3x_3 + 2x_4 \le 6, \tag{26}$$

which defines a facet of PS. □

Another consequence of Proposition 11 is

Theorem 2 *The inequality*

$$\sum_{j=1}^{n} max\{a_j, b - \sum_{j=1}^{K-1} a_j\} x_j \le b \tag{27}$$

defines a facet of PS.

Proof When $\sum_{j=1}^{K-1} a_j + a_n \ge b$, (27) and (1) coincide, and from Proposition 2, (27) is facet-defining. When $\sum_{j=1}^{K-1} a_j + a_n < b$, from Proposition 11,

$$\sum_{j=1}^{K} a_j x_j + (b - \sum_{j=1}^{K-1} a_j) x_n \le b \tag{28}$$

defines a facet of $PS(N - \{1, \dots, K, n\}, \emptyset)$. We now lift (28) with respect to $x_j, j \in N - \{1, \dots, K, n\}$, in any order. Let $l \in N - \{1, \dots, K, n\}$, and suppose we lift (28) with respect to x_l first. The lifting coefficient is

$$\alpha_l = \min\{\frac{b - \sum_{j=1}^{K} a_j x_j - (b - \sum_{i=1}^{K-1} a_i) x_n}{x_l} : x \in V(PS(N - \{1, \dots, K, n, l\}, \emptyset)) \text{ and } x_l > 0\}. \tag{29}$$

If $\max\{a_l, b - \sum_{j=1}^{K-1} a_j\} = b - \sum_{j=1}^{K-1} a_j$, then any $K-1$ variables among $x_1, \dots, x_K$ and x_n can be positive when x_l is positive. Because of Assumption 1. in Section 1, $\hat{x}$ given by

$$\hat{x}_j = \begin{cases} 1 & \text{if } j \in \{1, \dots, K-1, l\} \\ 0 & \text{otherwise} \end{cases}$$

is an optimal solution to (29), and so $\alpha_l = b - \sum_{j=1}^{K-1} a_j$.

Suppose now that $\max\{a_l, b - \sum_{j=1}^{K-1} a_j\} = a_l$. Let x^* be an optimal solution to (29). Assume that $x_n^* > 0$. Because $\sum_{j=1}^{K-2} a_j + a_l + a_n < b$, by repeating the argument above, the optimal value of (29) is $\alpha_l = b - \sum_{j=1}^{K-2} a_j - a_n$, in which case, $\alpha_l > a_l$. Assume now that $x_n^* = 0$. Since $x_l^* > 0$, one of $x_1^*, \dots, x_K^*$ must be 0. Suppose that $x_r^* = 0$, where $r \in \{1, \dots, K\}$. If $\sum_{j=1}^{r-1} a_j + \sum_{j=r+1}^{K} a_j + a_l < b$, then $\alpha_l = b - \sum_{j=1}^{r-1} a_j - \sum_{j=r+1}^{K} a_j > a_l$. If $\sum_{j=1}^{r-1} a_j + \sum_{j=r+1}^{K} a_j + a_l \geq b$, then clearly $\sum_{j=1}^{r-1} a_j x_j^* + \sum_{j=r+1}^{K} a_j x_j^* + a_l x_l^* = b$, which gives $\alpha_l = a_l$. Now, since $\sum_{j=1}^{K-1} a_j + a_l \geq b$, then $\alpha_l = a_l$.

Therefore, $\alpha_l = \max\{a_l, b - \sum_{j=1}^{K-1} a_j\}$. By using a similar argument, it can be shown that $\alpha_i = \max\{a_i, b - \sum_{j=1}^{K-1} a_j\}\ \forall i \in N - \{1, \dots, K, n, l\}$. □

Example 2 (Continued) Inequality (11) is not facet-defining, and it can be replaced with

$$4x_1 + 3x_2 + 2x_3 + 2x_4 + 2x_5 \leq 6.$$

□

We proved in Proposition 6 that (10) is facet-defining only if $a_1 + \sum_{j=n-K+2}^{n} a_j \leq b$. We next present a family of facet-defining inequalities that are stronger than (10) when $a_1 + \sum_{j=n-K+2}^{n} a_j > b$. The family of inequalities is a subclass of the following family of valid inequalities.

Lemma 2 *Let $U = \{u_1, \dots, u_{K-1}\}$ be a subset of N such that $a_{u_1} \geq \dots \geq a_{u_{K-1}}$ and $\sum_{j \in U} a_j < b$. Let $r \in N - U$ be such that $a_r + \sum_{j \in U} a_j > b$. Let $\{p, q\} \subseteq N - (U \cup \{r\})$ be such that $a_p \geq a_{u_1}$, $a_q \geq a_{u_1}$, and*

$$a_p + a_q + \sum_{j \in U - \{u_{K-1}\}} a_j \leq b. \tag{30}$$

Let T be a subset of $N - (U \cup \{r, p, q\})$ with $a_j \geq a_{u_1}\ \forall j \in T$. Then,

$$a_r x_r + \sum_{j\in T} max\{a_j, b - \sum_{i\in U} a_i\} x_j + (b - \sum_{i\in U} a_i) \sum_{j\in U\cup\{p,q\}} x_j \leq K(b - \sum_{j\in U} a_j) \quad (31)$$

defines a facet of $PS(N - (U \cup T \cup \{r,p,q\}), \emptyset)$.

Proof We start with

$$a_r x_r \leq b - \sum_{j\in U} a_j, \quad (32)$$

which is a cover inequality for $PS(N - (U \cup \{r\}), U)$. We first lift (32) with respect to x_p, x_q, and x_i $\forall i \in T$ in this order. The lifting coefficient of x_p is the greatest value of α_p for which

$$a_r x_r + \alpha_p x_p \leq b - \sum_{j\in U} a_j \quad (33)$$

$\forall x \in V(PS(N - (U \cup \{p,r\}), U))$. When $x_p = 0$ there is no restriction to the value of α_p. So suppose $x_p > 0$. Since $K-1$ variables are fixed at 1 in $PS(N-(U\cup\{p,r\}), U)$, when $x_p > 0$, $x_r = 0$. Because of (30) and $a_q \geq a_{u_1}$, $a_p + \sum_{j\in U} a_j \leq b$. This means that the greatest possible positive value of x_p when $x_j = 1$ $\forall j \in U$ is 1, and therefore, $\alpha_p = b - \sum_{j\in U} a_j$. Likewise, $\alpha_q = b - \sum_{j\in U} a_j$, and $\alpha_i = b - \sum_{j\in U} a_j$ $\forall i \in T$ such that $a_i + \sum_{j\in U} a_j \leq b$. For $i \in T$ such that $a_i + \sum_{j\in U} a_j > b$, the greatest possible positive value of x_i when $x_j = 1$ $\forall j \in U$ is $\frac{b - \sum_{j\in U} a_j}{a_i}$, and therefore $\alpha_i = a_i$. Thus,

$$a_r x_r + \sum_{j\in T} \max\{a_j, b - \sum_{i\in U} a_i\} x_j + (b - \sum_{j\in U} a_j)(x_p + x_q) \leq b - \sum_{j\in U} a_j \quad (34)$$

defines a facet of $PS(N - (U \cup T \cup \{r,p,q\}), U)$ (the lifting order of x_p, x_q, and $x_i, i \in T$, is actually irrelevant.)

Next we lift (34) with respect to $x_{u_1}, \ldots, x_{u_{K-1}}$ in this order. The lifting coefficient of x_{u_1} is

$$\alpha_{u_1} = \max\{\frac{a_r x_r + \sum_{j\in T} \max\{a_j, b - \sum_{i\in U} a_i\} x_j + (b - \sum_{j\in U} a_j)(x_p + x_q) - (b - \sum_{j\in U} a_j)}{1 - x_{u_1}} :$$
$$x \in V(PS(N - (U \cup T \cup \{r,p,q\}), \{u_2, \ldots, u_{K-1}\})) \text{ and } x_{u_1} < 1\}. \quad (35)$$

Let $\hat{x}$ be given by

$$\hat{x}_j = \begin{cases} 1 & \text{if } j \in (U - \{u_1\}) \cup \{p,q\} \\ 0 & \text{otherwise.} \end{cases}$$

Since $\hat{x} \in V(PS(N - (U \cup T \cup \{r,p,q\}), \{u_2, \ldots, u_{K-1}\}))$, $\hat{x}_{u_1} < 1$, and

$$\frac{a_r \hat{x}_r + \sum_{j\in T} \max\{a_j, b - \sum_{i\in U} a_i\} \hat{x}_j + (b - \sum_{j\in U} a_j)(\hat{x}_p + \hat{x}_q) - (b - \sum_{j\in U} a_j)}{1 - \hat{x}_{u_1}} = b - \sum_{j\in U} a_j,$$

it follows that

$$\alpha_{u_1} \geq b - \sum_{j \in U} a_j. \tag{36}$$

Let $x^* \in V(PS(N - (U \cup T \cup \{r, p, q\}), \{u_2, \ldots, u_{K-1}\}))$ be such that $x^*_{u_1} < 1$ and

$$\alpha_{u_1} = \frac{a_r x^*_r + \sum_{j \in T} \max\{a_j, b - \sum_{i \in U} a_i\} x^*_j + (b - \sum_{j \in U} a_j)(x^*_p + x^*_q) - (b - \sum_{j \in U} a_j)}{1 - x^*_{u_1}}.$$

Suppose that $x^*_{u_1} > 0$. Then, since the $K-1$ variables $x_j, j \in U$, are positive in x^*, at most one of the variables $x_j, j \in T \cup \{r, p, q\}$, can be positive in x^*. If $x^*_j = 0 \ \forall j \in T \cup \{r, p, q\}$,

$$\alpha_{u_1} = \frac{-(b - \sum_{j \in U} a_j)}{1 - x^*_{u_1}} < 0,$$

which is inconsistent with (36). Let $F = \{j \in T : a_j > b - \sum_{i \in U} a_i\} \cup \{r\}$. Because $a_s + \sum_{j \in U} a_j \leq b$ when $s \in \{p, q\} \cup (T - F)$, (1) cannot be satisfied at equality in case $x^*_s > 0$ and $x^*_{u_1} < 1$. Since we are assuming that $x^*_{u_1}$ is fractional, from Proposition 3 x^* can be a vertex of $PS(N - (U \cup T \cup \{r, p, q\}), \{u_2, \ldots, u_{K-1}\})$ only if it satisfies (1) at equality. Thus, we must have $x^*_s = 0 \ \forall s \in \{p, q\} \cup (T - F)$, and $x^*_l > 0$ for some $l \in F$. From Proposition 3, x^* cannot have more than one fractional component. Therefore, $x^*_l = 1$ and

$$x^*_{u_1} = \frac{b - a_l - \sum_{j \in U - \{u_1\}} a_j}{a_{u_1}},$$

or $\alpha_{u_1} = a_{u_1} \leq b - \sum_{j \in U} a_j$. So, it is not possible to obtain a solution to (35) better than $\hat{x}$ when $x^*_{u_1} > 0$.

Suppose now that $x^*_{u_1} = 0$, i.e.

$$\begin{aligned} \alpha_{u_1} = -(b - \sum_{j \in U} a_j) + \max\{ & a_r x_r + \sum_{j \in T} \max\{a_j, b - \sum_{i \in U} a_i\} x_j + (b - \sum_{j \in U} a_j)(x_p + x_q) : \\ & x \in V(PS(N - (U \cup T \cup \{r, p, q\}), \{u_2, \ldots, u_{K-1}\})) \text{ and } x_{u_1} = 0\}. \end{aligned} \tag{37}$$

If $x_i = 0 \ \forall i \in F$ in an optimal solution to (37), then since the objective function coefficients of $x_i, i \in \{p, q\} \cup (T - F)$, in (37) are all equal to $b - \sum_{j \in U} a_j$, and (30) implies that two of these variables can be positive and equal to 1, then $\alpha_{u_1} = b - \sum_{j \in U} a_j$. Finally, assume that $x_l > 0$ for some $l \in F$ in an optimal solution to (37). Because

$$\frac{b - \sum_{j \in U} a_j}{a_t} \geq \frac{a_l}{a_l}$$

and $a_l + a_i + \sum_{j \in U - \{u_1\}} a_j > b \ \forall i \in (T - F) \cup \{p, q\}$, it follows from Proposition 1 that for some $v \in (T - F) \cup \{p, q\}$ (37) has an optimal solution with $x_v = 1$ and $x_l = \frac{b - \sum_{j \in U - \{u_1\}} a_j - a_v}{a_l}$, which gives $\alpha_{u_1} = (b - \sum_{j \in U} a_j) - (a_v - a_{u_1}) \leq b - \sum_{j \in U} a_j$. In any case, $\alpha_{u_1} \leq b - \sum_{j \in U} a_j$, and therefore $\alpha_{u_1} = b - \sum_{j \in U} a_j$. By repeating this same argument, it can be shown that $\alpha_{u_2} = \cdots = \alpha_{u_{K-1}} = b - \sum_{j \in U} a_j$ (the lifting order of $x_j, j \in U$, is actually irrelevant). □

As a consequence of Lemma 2, we have that

Theorem 3 *Suppose* $\sum_{j=n-K+2}^{n} a_j < b$, $\sum_{j=n-K}^{n-1} a_j \leq b$, *and* $a_1 + \sum_{j=n-K+2}^{n} a_j > b$. *Then,*

$$a_1x_1 + \sum_{j=2}^{n-K-1} max\{a_j, b - \sum_{i=n-K+2}^{n} a_i\}x_j + (b - \sum_{i=n-K+2}^{n} a_i) \sum_{j=n-K}^{n} x_j \leq K(b - \sum_{j=n-K+2}^{n} a_j) \quad (38)$$

defines a facet of PS.

Proof $U = \{n-K+2, \ldots, n\}, r = 1, p = n-K, q = n-K+1$, and $T = \{2, \ldots, n-K-1\}$. $\square$

Note that when $a_1 + \sum_{j=n-K+2}^{n} a_j > b$, (38) is stronger than (10).

Example 3 Let $n = 4, K = 2$, and (1) be

$$6x_1 + 3x_2 + 2x_3 + x_4 \leq 6.$$

Then,

$$6x_1 + 5x_2 + 5x_3 + 5x_4 \leq 10 \quad (39)$$

defines a facet of *PS*. Inequality (39) is stronger than $\sum_{j=1}^{4} x_j \leq 2$. $\square$

Next, we present a family of valid inequalities that under certain conditions is facet-defining for *PS*. The family of inequalities is particularly useful when the constraint matrix is not dense, a situation that is common in applications.

Proposition 12 *Let* C, N_0, *and* N_1 *be three disjoint subsets of* N *with* $N = C \cup N_0 \cup N_1$ *and* $|C| = K - |N_1|$. *Assume that* C *is a cover for* $PS(N_0, N_1)$, *and*

$$a_p = min\{a_j : j \in C\} \quad (40)$$

(in case there is more than one knapsack coefficient satisfying (40), choose one arbitrarily). Suppose that

$$\sum_{j \in C - \{p\}} a_j < b - \sum_{j \in N_1} a_j, \quad (41)$$

and $a_l = 0$ *for some* $l \in N_0$. *Then,*

$$\sum_{j \in C} a_j x_j + \Delta \sum_{j \in N_0} x_j + \sum_{j \in N_1} \alpha_j x_j \leq b + \sum_{j \in N_1} (\alpha_j - a_j) \quad (42)$$

is valid for PS, where $\Delta = b - \sum_{j \in C - \{p\}} a_j - \sum_{j \in N_1} a_j$, *and*

$$\alpha_j = \begin{cases} \Delta + a_j & \text{if } a_p > \Delta + a_j \\ max\{a_p, a_j\} & \text{otherwise.} \end{cases} \quad (43)$$

Proof We prove the proposition by lifting $\sum_{j\in C} a_j x_j \leq b - \sum_{j\in N_1} a_j$ with respect to x_l, $x_j, j \in N_1$, and $x_j, j \in N_0 - \{l\}$, in this order. From Proposition 11,

$$\sum_{j\in C} a_j x_j + \Delta x_l \leq b - \sum_{j\in N_1} a_j \tag{44}$$

is facet-defining for $PS(N_0 - \{l\}, N_1)$.

We now lift (44) with respect to $x_j, j \in N_1$. Let $r \in N_1$, and suppose we lift (44) with respect to x_r first. The lifting coefficient, α_r, is

$$\alpha_r = \max\{\frac{\sum_{j\in C} a_j x_j + \Delta x_l - (b - \sum_{j\in N_1} a_j)}{1 - x_r} : x \in V(PS(N_0 - \{l\}, N_1 - \{r\})) \text{ and } x_r < 1\}. \tag{45}$$

Let x^* be an optimal solution to (45). Suppose $a_p > \Delta + a_r$. Then,

$$\sum_{j\in C} a_j > b - \sum_{j\in N_1} a_j + a_r. \tag{46}$$

If $x^*_r = 0$,

$$\alpha_r = \sum_{j\in C} a_j x^*_j + \Delta x^*_l - (b - \sum_{j\in N_1} a_j).$$

Because $|C| + |N_1| = K$ and $x^*_r = 0$, it is possible to satisfy (2) and have all components $x^*_j, j \in C$, and x^*_l positive. Since $a_l = 0$, clearly $x^*_l = 1$. On the other hand,

$$\sum_{j\in C} a_j x_j + a_r x_r \leq b - \sum_{j\in N_1} a_j + a_r \ \forall x \in V(PS(N_0 - \{l\}, N_1 - \{r\})),$$

and (46) imply that $\sum_{j\in C} a_j x^*_j = b - \sum_{j\in N_1} a_j + a_r$, and therefore,

$$\alpha_r = \Delta + a_r.$$

Assume now that $x^*_r > 0$. Since $x^*_r < 1$, x^*_r is fractional, and from Proposition 3,

$$\sum_{j\in C} a_j x^*_j + a_r x^*_r = b - \sum_{j\in N_1} a_j + a_r. \tag{47}$$

Because of (41), (47) can be satisfied only if $x^*_j > 0 \ \forall j \in C$. Since x^* cannot have more than one fractional component, $x^*_j = 1 \ \forall j \in C$. However, this is not possible when (46) holds. Therefore, $x^*_r = 0$, and so $a_p > \Delta + a_r \Rightarrow \alpha_r = \Delta + a_r$.

Suppose now $a_p \leq \Delta + a_r$, i.e. $\sum_{j\in C} a_j \leq b - \sum_{j\in N_1} a_j + a_r$. If $x^*_r = 0$,

$$\alpha_r = \sum_{j\in C} a_j + \Delta - (b - \sum_{j\in N_1} a_j) = a_p.$$

If $x_r^* > 0$, by repeating the argument for the case $a_p > \Delta + a_r$ and $x_r^* > 0$, we obtain $x_j^* = 1$ $\forall j \in C$, which, together with (2), implies that $x_l^* = 0$, and

$$\alpha_r = \frac{\sum_{j \in C} a_j - b + \sum_{j \in N_1} a_j}{1 - \frac{b - \sum_{j \in N_1 - \{r\}} a_j - \sum_{j \in C} a_j}{a_r}} = a_r.$$

Therefore, $a_p \leq \Delta + a_r \Rightarrow \alpha_r = \max\{a_p, a_r\}$. By an argument similar to the one used to calculate α_r, it can be shown that α_j satisfies (43) for all other $j \in N_1 - \{r\}$.

Next, we lift

$$\sum_{j \in C} a_j x_j + \Delta x_l + \sum_{j \in N_1} \alpha_j x_j \leq b + \sum_{j \in N_1} (\alpha_j - a_j) \tag{48}$$

with respect to $x_j, j \in N_0 - \{l\}$. First we lift (48) with respect to the variables x_j with $a_j = 0$, and we show that their lifting coefficients are $\alpha_j = \Delta$. Note that $a_p > \Delta$, and therefore $a_j = 0 \Rightarrow a_p > \Delta + a_j$.

Let $T \subset N_0 - \{l\}$ be such that $a_j = 0$ $\forall j \in T$, and suppose

$$\sum_{j \in C} a_j x_j + \Delta x_l + \Delta \sum_{j \in T} x_j + \sum_{j \in N_1} \alpha_j x_j \leq b + \sum_{j \in N_1} (\alpha_j - a_j) \tag{49}$$

defines a facet for $PS(N_0 - (T \cup \{l\}), \emptyset)$. Let $t \in N_0 - (T \cup \{l\})$. We lift (49) next with respect to x_t. Let $\hat{x}$ be an optimal solution to

$$\alpha_t = \min\{\frac{b - \sum_{j \in N_1} a_j + \sum_{j \in N_1} \alpha_j - \sum_{j \in C} a_j x_j - \Delta x_l - \Delta \sum_{j \in T} x_j - \sum_{j \in N_1} \alpha_j x_j}{x_t} :$$
$$x \in V(PS(N_0 - (T \cup \{l, t\}), \emptyset)) \text{ and } x_t > 0\}. \tag{50}$$

Since $a_t = 0$, we can assume that $\hat{x}_t = 1$. Also, because $\tilde{x}$ given by

$$\tilde{x}_j = \begin{cases} 1 & \text{if } j \in (C - \{p\}) \cup N_1 \cup \{t\} \\ 0 & \text{otherwise} \end{cases}$$

is a feasible solution to (50) with objective function value Δ,

$$\alpha_t \leq \Delta. \tag{51}$$

Suppose $\hat{x}_j > 0$ $\forall j \in T \cup \{l\}$. Because $a_j = 0$ $\forall j \in T \cup \{l\}$, $\hat{x}_j = 1$ $\forall j \in T \cup \{l\}$. Therefore,

$$\alpha_t = b - \sum_{j \in N_1} a_j + \sum_{j \in N_1} \alpha_j - \sum_{j \in C} a_j \hat{x}_j - \Delta(|T| + 1) - \sum_{j \in N_1} \alpha_j \hat{x}_j.$$

Let $C' = \{j \in C : \hat{x}_j > 0\}$ and $N_1' = \{j \in N_1 : \hat{x}_j > 0\}$. Since $x_l > 0$ and $x_t > 0$, $|N_1'| + |C'| + |T| \leq K - 2$. Because $|N_1| + |C| = K$,

$$|N_1 - N_1'| + |C - C'| \geq |T| + 2. \tag{52}$$

Assume now that $\sum_{j\in C} a_j\hat{x}_j + \sum_{j\in N_1} a_j\hat{x}_j < b$. From Proposition 3, $\hat{x}_j \in \{0,1\}$ $\forall j \in C \cup N_1$. It follows that,

$$\alpha_t = b - \sum_{j\in N_1} a_j + \sum_{j\in N_1-N_1'} \alpha_j - \sum_{j\in C'} a_j - \Delta(|T|+1) =$$

$$b - \sum_{j\in N_1} a_j + \sum_{j\in N_1-N_1'} \alpha_j - \sum_{j\in C'} a_j - b + \sum_{j\in C} a_j - a_p + \sum_{j\in N_1} a_j - \Delta|T| = \sum_{j\in N_1-N_1'} \alpha_j + \sum_{j\in C-C'} a_j - a_p - \Delta|T|.$$

Note that $a_j \geq a_p$ $\forall j \in C$, and that since $a_p > \Delta$, $\alpha_j \geq \Delta$ $\forall j \in N_1$. Thus, if $C - C' \neq \emptyset$ or $\alpha_j = \max\{a_p, a_j\}$ for some $j \in N_1 - N_1'$, $\alpha_t \geq \Delta(|T|+1) - \Delta|T| = \Delta$. If, on the other hand, $C - C' = \emptyset$ and $\alpha_j = a_j + \Delta$ $\forall j \in N_1 - N_1'$,

$$\sum_{j\in N_1-N_1'} \alpha_j + \sum_{j\in C-C'} a_j - a_p - \Delta|T| = \sum_{j\in N_1-N_1'} a_j + \Delta + \Delta(|N_1 - N_1'| - 1) - a_p - \Delta|T| =$$

$$\sum_{j\in N_1-N_1'} a_j + b - \sum_{j\in C} a_j + a_p - \sum_{j\in N_1} a_j + \Delta(|N_1 - N_1'| - 1) - a_p - \Delta|T| \geq$$

$$b - \sum_{j\in C} a_j - \sum_{j\in N_1'} a_j + \Delta(|T|+1) - \Delta|T| > \Delta,$$

where the last inequality follows from $\sum_{j\in C'} a_j + \sum_{j\in N_1'} a_j < b$ and $C = C'$. Now assume that $\sum_{j\in C} a_j\hat{x}_j + \sum_{j\in N_1} a_j\hat{x}_j = b$. Because of (41), $C - C' = \emptyset$, and from Proposition 1 we may assume that $\hat{x}_j = 1$ $\forall j \in N_1'$. Thus,

$$\alpha_t = b - \sum_{j\in N_1} a_j + \sum_{j\in N_1} \alpha_j - b + \sum_{j\in N_1} a_j\hat{x}_j - \Delta(|T|+1) - \sum_{j\in N_1} \alpha_j\hat{x}_j =$$
$$\sum_{j\in N_1-N_1'} (\alpha_j - a_j) - \Delta(|T|+1). \tag{53}$$

Because $C' = C$, $|N_1 - N_1'| \geq 2$. So let $u \in N_1 - N_1'$. Then,

$$\sum_{j\in C} a_j\hat{x}_j + \sum_{j\in N_1'} a_j = b \Rightarrow \sum_{j\in C} a_j + \sum_{j\in N_1'} a_j \geq b \Rightarrow$$

$$\sum_{j\in C} a_j + \sum_{j\in N_1'} a_j + a_u \geq b + a_u \Rightarrow \sum_{j\in C} a_j + \sum_{j\in N_1'} a_j + a_u + \sum_{j\in(N_1-N_1')-\{u\}} a_j \geq b + a_u \Rightarrow$$

$$\sum_{j\in C} a_j \geq b - \sum_{j\in N_1} a_j + a_u \Rightarrow a_p \geq \Delta + a_u \Rightarrow \alpha_u = \Delta + a_u.$$

Thus, $\alpha_j = a_j + \Delta$ $\forall j \in N_1 - N_1'$, and therefore, (53) implies that

$$\alpha_t = |N_1 - N_1'|\Delta - \Delta(|T|+1) \geq \Delta,$$

where the last inequality follows from (52) and $C = C'$. Thus, if $\hat{x}_j > 0\ \forall j \in T \cup \{l\}$, the objective function of (50) is not better than Δ.

Suppose $\hat{x}_v = 0$ for some $v \in T \cup \{l\}$. Because $a_v = 0$, x' given by

$$x'_j = \begin{cases} \hat{x}_j & \text{if } j \neq v \text{ and } j \neq t \\ \hat{x}_t & \text{if } j = v \\ 0 & \text{if } j = t \end{cases}$$

belongs to $PS(N_0 - (T \cup \{l, t\}), \emptyset)$. Thus,

$$\sum_{j\in C} a_j x'_j + \Delta x'_l + \Delta \sum_{j\in T} x'_j + \sum_{j\in N_1} \alpha_j x'_j + \alpha_t x'_t \leq b + \sum_{j\in N_1} (\alpha_j - a_j), \tag{54}$$

and

$$\sum_{j\in C} a_j \hat{x}_j + \Delta \hat{x}_l + \Delta \sum_{j\in T} \hat{x}_j + \sum_{j\in N_1} \alpha_j \hat{x}_j + \alpha_t \hat{x}_t = b + \sum_{j\in N_1} (\alpha_j - a_j). \tag{55}$$

Subtracting (55) from (54), we obtain $\alpha_t \geq \Delta$. Thus, $\alpha_j = \Delta\ \forall j \in N_0 - \{l\}$ with $a_j = 0$.

Finally, let $R = \{j \in N_0 - \{l\} : a_j = 0\}$. We now lift

$$\sum_{j\in C} a_j x_j + \Delta x_l + \Delta \sum_{j\in R} x_j + \sum_{j\in N_1} \alpha_j x_j \leq b + \sum_{j\in N_1} (\alpha_j - a_j) \tag{56}$$

with respect to x_j for $j \in N_0 - (R \cup \{l\})$, and we we show that the lifting coefficients are greater or equal to Δ as follows. First note that if $i, j \in N$ with $a_i > 0$ and $a_j = 0$, and $\tilde{x}$ is a feasible solution with $\tilde{x}_i > 0$ and $\tilde{x}_j = 0$, there is another feasible solution $\tilde{x}'$ that is equal to $\tilde{x}$, except that $\tilde{x}'_i = 0$ and $\tilde{x}'_j = \tilde{x}_i$. If $\sum_{j\in N} \alpha_j x_j \leq \alpha_0$ is a valid inequality for PS that is satisfied at equality by $\tilde{x}$, then $\alpha_i \geq \alpha_j$. In our case, $\alpha_j = \Delta$. Because $\tilde{x}$ may not exist, we consider a higher dimensional polytope in which $\tilde{x}$ exists, and we use Proposition 10 to establish the claim.

So, consider the polytope

$$\tilde{PS} = \text{conv}(\{x \in [0,1]^{n+K-1} : \sum_{j=1}^{n} a_j x_j \leq b \text{ and at most } K \text{ variables can be positive}\}).$$

Clearly PS and $\tilde{PS}(\{x_{n+1}, \ldots, x_{n+K-1}\}, \emptyset)$ are isomorphic. Also, it is clear that (56) is facet-defining for $\tilde{PS}(\{x_{n+1}, \ldots, x_{n+K-1}\} \cup N_0 - (R \cup \{l\}), \emptyset)$, and if we lift (56) with respect to $x_{n+1}, \ldots, x_{n+K-1}$, we obtain

$$\sum_{j\in C} a_j x_j + \Delta x_l + \Delta \sum_{j\in R} x_j + \sum_{j\in N_1} \alpha_j x_j + \Delta \sum_{j=n+1}^{n+K-1} x_j \leq b + \sum_{j\in N_1} (\alpha_j - a_j), \tag{57}$$

which is facet-defining for $\tilde{PS}(N_0 - (R \cup \{l\}), \emptyset)$. Let α_j be the lifting coefficient of x_j $\forall j \in N_0 - (R \cup \{l\})$. Let $g \in N_0 - (R \cup \{l\})$, and $x^{(1)} \in \tilde{PS}$ be such that $x^{(1)}_g > 0$ and it satisfies

$$\sum_{j\in C} a_jx_j + \Delta x_l + \Delta \sum_{j\in R} x_j + \sum_{j\in N_1} \alpha_j x_j + \Delta \sum_{j=n+1}^{n+K-1} x_j + \sum_{j\in N_0-(R\cup\{l\})} \alpha_j x_j \le b + \sum_{j\in N_1} (\alpha_j - a_j), \quad (58)$$

at equality. Because at most K variables can be positive, one of $x_j^{(1)}, j \in R \cup \{n+1, \ldots, n+K-1, l\}$ must be 0. Let $h \in R \cup \{n+1, \ldots, n+K-1, l\}$, and assume that $x_h^{(1)} = 0$. Now, let $x^{(2)} \in \tilde{PS}$ be given by

$$x_j^{(2)} = \begin{cases} x_j^{(1)} & \text{if } j \ne g \text{ or } j \ne h \\ x_g^{(1)} & \text{if } j = h \\ 0 & \text{if } j = g. \end{cases}$$

Since $x^{(2)}$ satisfies (58), $\alpha_g \ge \Delta$. Thus,

$$\alpha_j \ge \Delta \;\forall j \in N_0 - (R \cup \{l\}). \quad (59)$$

From Proposition 10, it follows that the lifting coefficient of $x_j, j \in N_0 - (R \cup \{l\})$ is greater or equal to Δ if (57) is lifted with respect to $x_j, j \in N_0 - (R \cup \{l\})$ before it is lifted with respect to $x_{n+1}, \ldots, x_{n+K-1}$. This means that (42) is valid for $\tilde{PS}(\{x_{n+1}, \ldots, x_{n+K-1}\}, \emptyset)$, and therefore it is valid for PS. □

As a consequence of Proposition 12, we have

Theorem 4 *If $a_j \le \Delta \;\forall j \in N_0$, (42) is facet-defining for PS.*

Proof Let α_j be the lifting coefficient of x_j for $j \in N_0$, as in the proof of Proposition 12. We know that $\alpha_j = \Delta \;\forall j \in N_0$ such that $a_j = 0$. Let $t \in N_0$ with $a_t > 0$. If $a_t \le \Delta$, then $\sum_{j\in C-\{p\}} a_j + \sum_{j\in N_1} a_j + a_t \le b$. This means that the point x^* given by

$$x_j^* = \begin{cases} 1 & \text{if } j \in (C - \{p\}) \cup N_1 \cup \{t\} \\ 0 & \text{otherwise} \end{cases}$$

belongs to PS, and therefore

$$\sum_{j\in C} a_j x_j^* + \sum_{j\in N_0} \alpha_j x_j^* + \sum_{j\in N_1} \alpha_j x_j^* \le b + \sum_{j\in N_1} (\alpha_j - a_j),$$

or,

$$\sum_{j\in C-\{p\}} a_j + \alpha_t + \sum_{j\in N_1} \alpha_j \le b + \sum_{j\in N_1} (\alpha_j - a_j),$$

and thus, $\alpha_t \le \Delta$. However, from (59), we know that $\alpha_t \ge \Delta$. So, $\alpha_t = \Delta$, and (42) is facet-defining for PS. □

Example 4 Let $n = 5$, $K = 3$, and (1) be given by

$$5x_1 + 5x_2 + 3x_3 + 0x_4 + 0x_5 \leq 9.$$

Then,

$$5x_1 + 5x_2 + 4x_3 + 4x_4 + 4x_5 \leq 13$$

is facet-defining for PS, where $C = \{1, 2\}$, $N_0 = \{3, 4\}$, and $N_1 = \{5\}$. □

4 Computational Experience

We tested the performance on difficult instances of CCOP of the

- MIP formulation, in which one introduces in the model auxiliary 0-1 variables, and models Constraint (2) with Constraints (5) and (6)
- continuous formulation, in which one keeps in the model only the continuous variables, and enforces Constraint (2) algorithmically through a branch-and-bound algorithm by using a specialized branching scheme
- continuous formulation through a branch-and-cut algorithm by using a specialized branching scheme and the lifted cover inequalities introduced in Section 3 as cuts.

The number of knapsack constraints, m, and the number of variables, n, in the instances of CCOP tested are given in the first column of Table 1. We tested 3 different instances for each pair $m \times n$. The cardinality of the instances, K, is given in the second column of Table 1. The last column of Table 1 gives the densities of the constraint matrices, which are equal to

$$100 \times \frac{\text{number of nonzero coefficients of the knapsack constraints}}{mn}.$$

The instances with same m and n had the same cardinality and density.

The instances were randomly generated as follows. The profit coefficients $c_j, j \in N$, were integers uniformly generated between 10 and 25. The knapsack coefficients, $a_{ij}, i \in M, j \in N$, were integers uniformly generated between 5 and 20. The $m \times$ density indices of the nonzero knapsack coefficients were uniformly generated between 1 and n. The right-hand-sides of the knapsack constraints were given by

$$b_i = \max\{\lfloor .3 \sum_{j \in N} a_{ij} \rfloor, \text{greatest coefficient of the } i^{\text{th}} \text{ knapsack} + 1\}, i \in M.$$

The values of K and of the densities of the coefficient matrices were chosen by selecting the hardest values determined by performing preliminary computational tests.

Table 1: Cardinality and density

$m \times n$	K	**% density**
20×500	150	50
$20 \times 1,000$	300	50
$20 \times 1,500$	450	50
$20 \times 2,000$	600	30
$20 \times 2,500$	750	42
$30 \times 3,000$	1,000	33
$30 \times 3,500$	1,000	28
$50 \times 4,000$	1,000	25
$50 \times 4,500$	2,000	30
$50 \times 5,000$	2,000	20
$50 \times 5,500$	2,000	18
$50 \times 6,000$	2,000	16
$50 \times 6,500$	1,000	15
$50 \times 7,000$	2,000	14
$70 \times 7,500$	2,000	13
$70 \times 8,000$	3,000	25

The continuous formulation was tested using MINTO 3.0 [23] with CPLEX 6.6 as LP solver. Our motivation for using MINTO was the flexibility that it offers to code alternative branching schemes, feasibility tests, and separation routines. Initially we implemented the MIP formulation with MINTO. However, MINTO proved to be too slow when compared to CPLEX 6.6 to solve the MIPs. Also, CPLEX 6.6 has Gomory cuts, which we thought could be helpful in reducing the effort required to complete the enumeration [2]. Ultimately, by using CPLEX 6.6 to run the MIPs, we wanted to give the MIP formulation its best chance, even though in principle this would be unfair to the continuous formulation.

4.1 Specialized Branching Scheme for the Continuous Formulation

We adopted the specialized branching scheme proposed by Bienstock in [6]. Suppose that more than K variables are positive in a solution $\tilde{x}$ of the LP relaxation of CCOP, and that $\tilde{x}_l$ is one of them. Then, we may divide the solution space by requiring in one branch, which we call *down*, that $x_l = 0$, and in the other branch, which we call *up*, that

$$\sum_{j \in N - \{l\}} x_j \leq K - 1. \tag{60}$$

Let $S_{\text{down}} = S \cap \{x \in \Re^n : x_l = 0\}$ and $S_{\text{up}} = S \cap \{x \in \Re^n : x \text{ satisfies (60)}\}$. Clearly

$S = S_{\text{down}} \cup S_{\text{up}}$, although it may happen that $\tilde{x} \in S_{\text{up}}$.

In general, suppose that at the current node of the enumeration tree the variables $x_j, j \in F$, are free, i.e. have not been branched on, and that t variables have been branched up. We branch at the current node on variable x_p by imposing in the down branch that $x_p = 0$, and in the up branch that

$$\sum_{j \in F - \{p\}} x_j \leq K - t - 1.$$

Even though the current LP relaxation solution may reoccur in the up branch, this branching scheme ends within a finite number of nodes, since we can fathom any node that corresponds to K up branches.

4.2 Cuts for the Continuous Formulation

Since each knapsack constraint of CCOP has a large number of 0 coefficients in the data we used to test CCOP, the conditions of Proposition 6 are satisfied in general, and (10) is facet-defining for the convex hull of the feasible sets of the instances tested. Because of this, we included (10) in the initial formulation (LP relaxation) of the instances of CCOP we tested, which considerably tightened the model.

Initially, we wanted to use (27) of Theorem 2 in a preprocessing phase to tighten any inequality in the initial formulation for which

$$a_{ij} < b_i - \sum_{l=1}^{K-1} a_{il} \tag{61}$$

for some $i \in M, j \in N$. However, we did not detect condition (61) in any of the instances we tested. Note that the specific way we defined $b_i, i \in M$, in the instances was such that the knapsack inequalities were already tight, in the sense that condition (61) is not satisfied. On the other hand, in real world instances, where many times models are not tight when defined, (27) might be useful.

We also wanted initially to use (38) to define the up branches every time it was stronger than (10), as would happen if the conditions of Theorem 3 were present in at least one of the knapsack constraints. However, because of the number of 0 coefficients in the data we tested, in general $a_{i1} + \sum_{j=n-K+2}^{n} a_{ij} \leq b_i, i \in M$.

We used (42) as a cut. Unlike (27) and (38), (42) proved to be very useful in the instances we tested, and therefore we used it in our branch-and-cut algorithm.

Given a point $\tilde{x}$ that belongs to the LP relaxation of CCOP and that does not satisfy (2), to find a violated inequality (42), we have to select disjoint subsets $C, N_1 \subset N$, such that $C \neq \emptyset$, $|C| + |N_1| = K$, $\sum_{j \in C} a_{ij} + \sum_{j \in N_1} a_{ij} > b_i$, $\sum_{j \in C - \{p\}} a_{ij} + \sum_{j \in N_1} a_{ij} < b_i$, with $a_{ip} = \min\{a_{ij} : j \in C\}$, $a_{ij} = 0$ for some $j \in N - (C \cup N_1)$ with $\tilde{x}_{ij} > 0$, and

$$\sum_{j \in C} a_{ij}\tilde{x}_j + \Delta \sum_{j \in N_0} \tilde{x}_j + \sum_{j \in N_1} \alpha_j \tilde{x}_j > b_i + \sum_{j \in N_1} (\alpha_j - a_{ij}), \tag{62}$$

where $N_0 = N - (C \cup N_1)$. It appears then the separation problem for (42) is difficult. Thus, we used a heuristic, which we now describe, to solve the separation problem.

Let $i \in M$ be such that $\sum_{j \in N} a_{ij}\tilde{x}_i = b_i$. Because the terms corresponding to $j \in C$ in (42) are $a_{ij}x_j$, we require that $a_{ij} > 0$ and $\tilde{x}_j > 0 \ \forall j \in C$. Because $j \in N_1$ contributes $\alpha_j x_j$ to the left-hand-side and α_j to the right-hand-side of (42), we require that $\tilde{x}_j = 1 \ \forall j \in N_1$. Since C cannot be empty, we first select its elements. We include in C as many $j \in N$ with $a_{ij} > 0$ and $\tilde{x}_j \in (0, 1)$ as possible, up to K elements. If $|C| = K$, we include in N_0 every $j \in N - C$ (in this case $N_1 = \emptyset$). If $|C| < K$, we include in N_1 all $j \in N - C$ with $\tilde{x}_j = 1$, in nonincreasing order of a_{ij} until $|C| + |N_1| = K$ or there are no more components of $\tilde{x}$ with $\tilde{x}_j = 1$. If $|C| + |N_1| < K$ or if there is no $j \in N - (C \cup N_1)$ with $a_{ij} = 0$ and $\tilde{x}_j > 0$, we fail to generate a cut for $\tilde{x}$ out of row i. If $|C| + |N_1| = K$ and $\exists j \in N - (C \cup N_1)$ with $a_{ij} = 0$ and $\tilde{x}_j > 0$, we make $N_0 = N - (C \cup N_1)$. Finally, if (62) holds, we succeed in generating a cut for $\tilde{x}$ out of row i. Otherwise, we fail.

The following example comes from our preliminary computational experience.

Example 5 Let $m = 2$, $n = 100$, and $K = 20$. The solution of the LP relaxation is $\tilde{x}$ given by $\tilde{x}_1 = \tilde{x}_{11} = \tilde{x}_{17} = \tilde{x}_{22} = \tilde{x}_{43} = \tilde{x}_{45} = \tilde{x}_{56} = \tilde{x}_{57} = \tilde{x}_{60} = \tilde{x}_{61} = \tilde{x}_{62} = \tilde{x}_{64} = \tilde{x}_{68} = \tilde{x}_{70} = \tilde{x}_{79} = \tilde{x}_{80} = \tilde{x}_{86} = \tilde{x}_{95} = 1$, $\tilde{x}_{19} = 0.434659$, $\tilde{x}_{75} = 0.909091$, $\tilde{x}_{78} = 0.65625$, and $\tilde{x}_j = 0$ otherwise. (Note that 3 variables are fractional even though $m = 2$. This is because (10) was included in the original formulation.) Let $N_0' = \{j \in N : \tilde{x}_j = 0\}$. Below we give the terms of one of the knapsack constraints for which $\tilde{x}_j > 0$ and $a_{ij} > 0$:

$$14x_1 + 30x_{11} + 32x_{19} + 20x_{22} + 12x_{45} + 14x_{56} + 12x_{61} + 24x_{70} + 32x_{75} + 24x_{79} \leq 193.$$

The 18 components of $\tilde{x}$ equal to 1 are all included in N_1, i.e. $N_1 = \{1, 11, 17, 22, 43, 45, 56, 57, 60, 61, 62, 64, 68, 70, 79, 80, 86, 95\}$. The sets $C = \{19, 75\}$ and $N_0 = \{78\} \cup N_0'$, $\Delta = 11$, and the inequality (42) is

$$32x_{19}+32x_{75}+11x_{78}+11\sum_{j \in N_0'} x_j+25x_1+32x_{11}+11x_{17}+31x_{22}+11x_{43}+23x_{45}+25x_{56}+11x_{57}+$$

$$11x_{60} + 23x_{61} + 11x_{62} + 11x_{64} + 11x_{68} + 32x_{70} + 32x_{79} + 11x_{80} + 11x_{86} + 11x_{95} \leq 376. \quad (63)$$

Since this inequality is violated by $\tilde{x}$, it is included in the formulation at the root node. Because we fail to generate a cut for $\tilde{x}$ out of the other knapsack inequality, this is the only cut included at this time, and we re-solve the LP relaxation with (63) included in the formulation.

□

We search for a cut (42) for $\tilde{x}$ from every knapsack inequality (7) and we include in the formulation at the current node as many cuts as we can find. In the case where we find cuts, we re-solve the LP relaxation with the cuts added. Otherwise, we branch. We search for cuts at every node of the enumeration tree.

4.3 Branch-and-Bound Alternatives

We performed preliminary tests with MINTO's pre-processing and node selection alternatives. The ones that performed best were: perform pre-processing and limited probing, and use best-bound node selection. We then used these options in our computation for the continuous formulation. For variable selection in Bienstock's branching scheme we used a *least index rule*, i.e. select for branching the positive variable with least index. We used the default options of CPLEX to test the MIP formulation.

4.4 Computational Results

We used a Sun Ultra 2 with two UltraSPARC 300 MHz CPUs and 256 MB memory to perform the computational tests. The results are summarized in Table 2. Table 2 gives, for each pair $m \times n$, the average number of nodes , CPU seconds, and number of cuts for the MIP and continuous formulations. The second column gives the average number of nodes over the 3 instances with the same m and n generated by CPLEX 6.6 to solve the MIP formulation to proven optimality. The next column, Cont. B&B, gives the average number of nodes generated by MINTO 3.0 to solve the continuous formulation exactly with Bienstock's branching scheme without the use of cuts, i.e. through a pure branch-and-bound approach. The following column, Cont. inc. (42), gives the number of nodes generated by MINTO 3.0 to solve the continuous formulation exactly with Bienstock's branching scheme and (42) as cuts, i.e. through a branch-and-cut approach. The column "% Red." gives the percentage reduction in number of nodes by using the continuous formulation and a branch-and-cut algorithm with (42) as cuts over the MIP formulation. Note that overall, the percentage reduction in number of nodes by using branch-and-cut over branch-and-bound for the continuous formulation was 70%. This means that the use of a branch-and-cut approach to solve the continuous formulation can be considerably more effective than pure branch-and-bound. The great overall reduction of 97% in the average number of nodes by using the continuous formulation with a branch-and-cut approach over the MIP approach indicates that by adding auxiliary 0-1 variables and enforcing (2) through their integrality does not take as much advantage of the combinatorial structure of the problem as in the continuous approach, where we fathom a node when (2) is satisfied, and where we branch by using Bienstock's scheme. As mentioned in Section 4.2, (10) is usually facet-defining, and thus using it to define the up branches may considerably help to reduce the upper bound on the up branches.

The four columns under "Time" in Table 2 have similar meanings to the four columns under "Nodes". The overall time reduction by using (42) as cuts in a branch-and-cut scheme to solve the continuous formulation over pure branch-and-bound was 62%, which indicates that a branch-and-cut approach may be considerably more efficient to solve the continuous formulation than branch-and-bound.

Because CPLEX 6.6 is much faster than MINTO 3.0, the overall time reduction of 78% of branch-and-cut on the continuous formulation over the MIP formulation is significant. We believe that such a great time reduction is not just the result of the reduction in the number of nodes, but also due to the fact that the size of the MIP formulation is 2 times greater than the continuous formulation, which becomes significant in the larger instances. Also, the degeneracy introduced by the variable upper bound constraints (5) may be harmful.

The only cuts generated by CPLEX 6.6 with default options were the Gomory cuts. The column Cuts Gomory at the end of Table 2 gives the average number of Gomory cuts generated by CPLEX 6.6. As we mentioned in Section 4, initially we tested the MIP approach with MINTO 3.0. MINTO generated a very large number of lifted flow cover inequalities (LFCIs), typically tens of thousands, even for the smallest instances. The reason is because the violation tolerance for LFCIs in MINTO is much smaller than in CPLEX. To verify the effectiveness of LFCIs in CPLEX we increased their violation tolerance. Our preliminary tests indicated that even though LFCIs help reduce the integrality gap faster, they are not effective in closing the gap, and we then kept CPLEX's default. The average number of Inequality (42) generated by MINTO is given in the last column of Table 2.

Table 2: Average number of nodes, time, and number of cuts for the MIP and the continuous formulations

$m \times n$	**Nodes**				**Time**				**Cuts**	
		Cont.		**%**		**Cont.**		**%**	**Gomory**	**(42)**
	MIP	**B&B**	**inc. (42)**	**Red.**	**MIP**	**B&B**	**inc. (42)**	**Red.**		
20×500	36,364	681	531	98	527	17	94	82	12	10
$20 \times 1,000$	109,587	1,360	315	99	2,141	792	208	90	75	23
$20 \times 1,500$	33,761	1,423	229	99	1,746	203	109	93	67	18
$20 \times 2,000$	12,738	2,753	729	94	802	839	318	60	153	72
$20 \times 2,500$	46,873	3,479	1,157	97	9,959	1,558	770	92	148	69
$30 \times 3,000$	196,010	3,927	1,092	99	36,288	3,570	720	98	205	162
$30 \times 3,500$	20,746	161	3	99	14,507	44	12	99	54	6
$50 \times 4,000$	18,529	289	112	99	14,798	57	49	99	97	11
$50 \times 4,500$	26,811	4,230	1,358	94	35,601	19,758	9,505	73	79	91
$50 \times 5,000$	39,776	5,553	1,749	95	53,388	23,570	11,320	78	161	211
$50 \times 5,500$	43,829	6,763	2,129	95	65,423	27,140	12,639	80	134	287
$50 \times 6,000$	49,574	7,981	2,727	94	72,751	28,923	13,547	81	231	320
$50 \times 6,500$	54,251	8,975	3,152	94	85,721	34,188	15,754	81	166	289
$50 \times 7,000$	17,524	163	158	99	24,439	577	715	97	61	15
$70 \times 7,500$	45,279	8,572	1,359	96	96,714	98,576	32,711	66	244	412
$70 \times 8,000$	32,497	9,085	2,168	93	89,238	106,684	31,730	64	315	601
TOTAL	784,149	65,395	18,968	97	604,043	346,496	130,201	78	2,202	2,597

5 Further Research

Given the encouraging computational results in this paper, it is important to study the following questions on branch-and-cut for CCOP:

- how can lifted cover inequalities be separated efficiently?
- how can cover inequalities be lifted efficiently in any order, either exactly or approximately, to obtain strong cuts valid for *PS*?
- in which order should cover inequalities be lifted?
- are there branching strategies more effective than Bienstock's [6]? (See [12] for an alternative branching strategy.)

Note that it is not possible to complement a variable x_j with $a_j < 0$, as it is usually done for the 0-1 knapsack problem, and keep the cardinality constraint (2) intact. This means that Assumption 2. of Section 1 implies a loss of generality. Thus, it is important to investigate the cardinality knapsack polytope when $a_j < 0$ for some of the knapsack coefficients.

Besides cardinality, there exists a small number of other combinatorial constraints, such as semi-continuous and SOS [12], that are pervasive in practical applications. We suggest investigating their polyhedral structure in the space of the continuous variables, and comparing the performance of branch-and-cut without auxiliary 0-1 variables for these problems against the usual MIP approach.

Recently, there has been much interest in bringing together the tools of integer programming (IP) and constraint programming (CP) [29] in a unified approach. Traditionally in IP, combinatorial constraints on continuous variables, such as cardinality, semi-continuous or SOS are modeled as mixed-integer programs (MIPs) by introducing auxiliary 0-1 variables and additional constraints. Because the number of variables and constraints becomes larger and the combinatorial structure is not used to advantage, these MIP models may not be solved satisfactorily, except for small instances. Traditionally, CP approaches to such problems keep and use the combinatorial structure, but do not use linear programming (LP) bounds. In the approach used in this paper the combinatorial structure of the problem is explored and no auxiliary 0-1 variables are introduced. Nevertheless, we used strong bounds based on LP relaxations. As continued research, we suggest the use of this approach in a combined IP/CP strategy to solve difficult instances of CCOP.

References

[1] E. Balas, "Facets of the Knapsack Polytope," *Mathematical Programming* 8, 146-164 (1975).

[2] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj, "Gomory Cuts Revisited," *Operations Research Letters* 19, 1-9 (1996).

[3] E.L.M. Beale, "Integer Programming," in: K. Schittkowski (Ed.), *Computational Mathematical Programming*, NATO ASI Series, Vol. F15, Springer-Verlag, 1985, pp. 1-24.

[4] E.L.M. Beale and J.A. Tomlin, "Special Facilities in a General Mathematical Programming System for Nonconvex Problems Using Ordered Sets of Variables," in: J. Lawrence (Ed.), *Proceedings of the fifth Int. Conf. on O.R.*, Tavistock Publications, 1970, pp. 447-454.

[5] L.T. Biegler, I.E. Grossmann, and A.W. Westerberg, *Systematic Methods of Chemical Process Design*, Prentice Hall, 1997.

[6] D. Bienstock, "Computational Study of a Family of Mixed-Integer Quadratic Programming Problems," *Mathematical Programming* 74, 121-140 (1996).

[7] H. Crowder, E.L. Johnson, and M.W. Padberg, "Solving Large Scale Zero-One Problems," *Operations Research* 31, 803-834 (1983).

[8] G.B. Dantzig, "On the significance of Solving Linear Programming Problems with some Integer Variables," *Econometrica* 28, 30-44 (1960).

[9] I.R. de Farias, Jr., "A Polyhedral Approach to Combinatorial Complementarity Programming Problems," *Ph.D. Thesis*, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA (1995).

[10] I.R. de Farias, Jr., "A Family of Facets for the Uncapacitated p-Median Polytope," *Operations Research Letters* 28, 161-167 (2001).

[11] I.R. de Farias, Jr., E.L. Johnson, and G.L. Nemhauser "A Generalized Assignment Problem with Special Ordered Sets: A Polyhedral Approach," *Mathematical Programming* 89, 187-203 (2000).

[12] I.R. de Farias, Jr., E.L. Johnson, and G.L. Nemhauser "Branch-and-Cut for Combinatorial Optimization Problems without Auxiliary Binary Variables," *Knowledge Engineering Review* 16, 25-39 (2001).

[13] I.R. de Farias, Jr., E.L. Johnson, and G.L. Nemhauser, "Facets of the Complementarity Knapsack Polytope," to appear in *Mathematics of Operations Research*.

[14] I.R. de Farias, Jr. and G.L. Nemhauser, "A Family of Inequalities for the Generalized Assignment Polytope," *Operations Research Letters* 29, 49-51 (2001).

[15] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, 1979.

[16] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh, "Lifted Flow Cover Inequalities for Mixed 0-1 Integer Programs," *Mathematical Programming* 85, 439-467 (1999).

[17] M. Grötschel, M. Jünger, and G. Reinelt, "A Cutting Plane Algorithm for the Linear Ordering Problem," *Operations Research* 32, 1195-1220 (1984).

[18] P.L. Hammer, E.L. Johnson, and U.N. Peled, "Facets of Regular 0-1 Polytopes," *Mathematical Programming* 8, 179-206 (1975).

[19] K. Hoffman and M.W. Padberg, "LP-Based Combinatorial Problem Solving," *Annals of Operations Research* 4, 145-194 (1985).

[20] J.N. Hooker, G. Ottosson, E.S. Thornsteinsson, and H.-J. Kim, "A Scheme for Unifying Optimization and Constraint Satisfaction Methods," *Knowledge Engineering Review* 15, 11-30 (2000).

[21] H. Marchand, A. Martin, R. Weismantel, and L.A. Wolsey, "Cutting Planes in Integer and Mixed-Integer Programming," CORE Discussion Paper (1999).

[22] H.M. Markowitz and A.S. Manne, "On the Solution of Discrete Programming Problems," *Econometrica* 25, 84-110 (1957).

[23] G.L. Nemhauser, G.C. Sigismondi, and M.W.P. Savelsbergh, "MINTO, a Mixed-INTeger Optimizer," *Operations Research Letters* 15, 47-58 (1994).

[24] G.L. Nemhauser and L.A. Wolsey, *Integer Programming and Combinatorial Optimization*, John Wiley and Sons, 1988.

[25] M.W. Padberg, "A Note on Zero-One Programming," *Operations Research* 23, 883-837 (1975).

[26] M.W. Padberg and G. Rinaldi, "Optimization of a 532-City Symmetric Traveling Salesman Problem by Branch-and-Cut," *Operations Research Letters* 6, 1-7 (1987).

[27] A.F. Perold, "Large-Scale Portfolio Optimization," *Management Science* 30, 1143-1160 (1984).

[28] R. Raman and I.E. Grossmann, "Symbolic Integration of Logic in MILP Branch-and-Bound Methods for the Synthesis of Process Networks," *Annals of Operations Research* 42, 169-191 (1993).

[29] P. van Hentenryck, *Constraint Satisfaction in Logic Programming*, MIT Press, 1989.

[30] T.J. van Roy and L.A. Wolsey, "Solving Mixed-Integer Programming Problems Using Automatic Reformulation," *Operations Research* 35, 45-57 (1987).

[31] L.A. Wolsey, "Faces for a Linear Inequality in 0-1 Variables," *Mathematical Programming*, 8, 165-178 (1975).

[32] L.A. Wolsey, "Facets and Strong Valid Inequalities for Integer Programs," *Operations Research* 24, 367-372 (1976).

[33] L.A. Wolsey, *Integer Programming*, John Wiley and Sons, 1998.