Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ALEXANDER MARTIN AND ROBERT WEISMANTEL

Conjunctive Cuts for Integer Programs

Preprint SC 98-18 (July 1998)

Conjunctive Cuts for Integer Programs

Alexander Martin Robert Weismantel*

July 6, 1998

Abstract

This paper deals with a family of conjunctive inequalities. Such inequalities are needed to describe the polyhedron associated with all the integer points that satisfy several knapsack constraints simultaneously. Here we demonstrate the strength and potential of conjunctive inequalities in connection with lifting from a computational point of view.

1 Introduction

The solution of general mixed integer programs is one of the challenging problems in discrete optimization. The problems that can be modeled as mixed integer programs arise in science, technology, business, and environment, and their number is tremendous and increasing. Such developments go side by side with the development of tools for tackling mixed integer programs. One important breakthrough that still drives the current activities in this field was the code of Crowder, Johnson, and Padberg [1983] that was able to solve real-world 0/1 integer programs that many researchers considered untracktable by this time. Their main idea was to interpret each single row of the constraint matrix as a knapsack constraint and to strengthen the original integer program by adding inequalities associated with relaxations in form of knapsack problems. An analysis of other important relaxations of an integer program allows to incorporate odd hole and clique inequalities for the stable set polyhedron (Padberg [1973]) or flow cover inequalities for certain mixed integer models (Padberg, Roy, and Wolsey [1985], Roy and Wolsey [1987]). Further recent examples of this approach are given in Ceria, Cordier, Marchand, and Wolsey [1998], Marchand and Wolsey [1997].

Instead of deriving combinatorial relaxation one may derive general cutting planes that do not exploit the underlying structure. Examples are (mixed) integer Chvátal-Gomory cuts or disjunctive cutting planes. The idea to derive disjunctive cutting planes is to split a polyhedron into two or more disjoint polyhedra, study these polyhedra individually and convexify the individual descriptions. In Balas, Ceria, and Cornuéjols [1993], [1996] it is shown that disjunctive cutting planes help solving general mixed integer programs.

We introduce here a somewhat contrary approach. Instead of breaking an integer program apart we study the convex hull of integer points that lie in the intersection of several polyhedra simultaneously. These polyhedra are deduced from the individual rows of the constraint system. In this respect, our approach is an extension of the idea of Crowder, Johnson and Padberg. Inequalities that are necessary to describe the "intersection polyhedron" by means of inequalities will be called *conjunctive cutting planes.*

*Supported by a "Gerhard Hess-Forschungsförderpreis" of the German Science Foundation (DFG).

We have incorporated a family of conjunctive cuts, called feasible set inequalities, in our general mixed integer programming solver. This algorithm includes tools for decomposing an integer program into blocks of moderate sizes. With each of these blocks there is associated a smaller integer program that may be viewed as the intersection of several knapsack constraints in general integer variables. In a companion paper we have studied this polyhedron by means of valid inequalities. Here we show how to turn this polyhedral knowledge into a practical separation algorithm. This separation algorithm has two major ingredients: One is a search procedure for feasible solutions of an integer program that we use to derive valid conjunctive inequalities associated with these feasible solutions. The other ingredient is an effective algorithm for lifting general integer variables, i. e., for turning low dimensional faces of an integer program into faces of higher dimension. Our computational results demonstrate the benefits and practical potential of this family of conjunctive cuts when used within a general mixed integer programming solver.

2 Feasible Set Inequalities: A Family of Conjunctive Cuts

Consider some finite sets $N, M \subset \mathbb{N}$, some matrix $A \in \mathbb{R}^{M \times N}$, vectors $b \in \mathbb{R}^M$, $u \in \mathbb{R}^N$, and the polytope

$$P\,(N, M, A, b, u) := \operatorname{conv}\{x \in \mathbb{Z}^N : \; Ax \leq b, \; 0 \leq x \leq u\},$$

that is the convex hull of all integral vectors x satisfying $Ax \leq b$ and $0 \leq x \leq u$.

Throughout this paper we use the following notation. We will assume $N = \{1, \dots, n\}$ and $M = \{1, \dots, m\}$ for $n, m \in \mathbb{N}$. For $v \in \mathbb{R}^N$, we denote by v^+ the vector with $v_i^+ = v_i$ if $v_i \geq 0$ and $v_i^+ = 0$, otherwise. Accordingly, v^- is the vector with $v_i^- = -v_i$ if $v_i \leq 0$ and $v_i^- = 0$, otherwise. If $v \in \mathbb{R}^N$, then $\operatorname{supp}(v)$ denotes the support of v; in formulas, $\operatorname{supp}(v) := \{i \in N : v_i \neq 0\}$. For $I \subseteq M$ and $J \subseteq N$, we denote by $A_{I,J} \in \mathbb{R}^{I \times J}$ the submatrix with entries $(a_{ij})_{i \in I, j \in J}$. We abbreviate $A_{I,N}$ by $A_{I \cdot}$ and $A_{M,J}$ by $A_{\cdot J}$. For $j \in N$, we write $A_{\cdot j}$ instead of $A_{\cdot \{j\}}$ to denote the j-th column of A. Accordingly, $A_{i \cdot}$ denotes the i-th row of A, for $i \in M$. For some vector $x \in \mathbb{R}^N$ and $S \subseteq N$, we denote by $x_S := (x_i)_{i \in S}$ the vector restricted to the components in S. We abbreviate $P\,(N, M, A, b, u)$ by P if it is clear from the context. For convenience, we also use the symbol $P\,(S)$ to abbreviate the polytope $P\,(S, M, A_{\cdot S}, b, u_S)$.

Definition 2.1 *Let $T \subseteq N$ such that $\sum_{i \in T} A_{\cdot i} v_i \leq b$ for all $v \in \mathbb{R}^T$ with $v \leq u_T$. T is called a* feasible set. *Let $w : T \mapsto \mathbb{Z}_+$ be some non-negative* weighting *of the elements of T. For $j \in N \setminus T$ with $\sum_{i \in T} A_{\cdot i} u_i + A_{\cdot j} u_j \not\leq b$, the inequality*

$$\sum_{i \in T} w_i x_i + w_j x_j \leq \sum_{i \in T} w_i u_i \tag{1}$$

is called a feasible set inequality associated with T (and $\{j\}$) *if*

$$w_j \;\; \leq \min_{l=1,\dots,u_j} \frac{1}{l} \; \min_x \;\; \begin{array}{l} \sum_{i \in T} w_i x_i \\ \sum_{i \in T} A_{\cdot i} x_i \geq A_{\cdot j} - r(T) \\ 0 \leq x_i \leq u_i, x_i \in \mathbb{Z}, i \in T, \end{array} \tag{2}$$

where $r(T) := b - \sum_{i \in T} A_{\cdot i} u_i$.

Proposition 2.2 *Feasible set inequalities are valid for $P(T \cup \{j\})$.*

Proof. Let $\gamma = \sum_{i \in T} w_i u_i$ and $\gamma_l := \max\{\sum_{i \in T} w_i x_i : \sum_{i \in T} A_{\cdot i} x_i + A_{\cdot j} l \leq b, 0 \leq x_i \leq u_i, x_i \in \mathbb{Z}, i \in T\}$. After complementing variables x_i to $u_i - x_i$ for $i \in T$ we obtain that the right-hand side of (2) is $\min_{l=1,\dots,u_j} \frac{1}{l}(\gamma - \gamma_l)$. For some integer solution $\bar{x} \in P(T \cup \{j\})$ with $\bar{x}_j \geq 1$ we have $\sum_{i \in T} w_i \bar{x}_i + w_j \bar{x}_j \leq \gamma_{\bar{x}_j} + \bar{x}_j \min_{l=1,\dots,u_j} \frac{1}{l}(\gamma - \gamma_l) \leq \gamma_{\bar{x}_j} + (\gamma - \gamma_{\bar{x}_j}) = \gamma$. In case $\bar{x}_j = 0$ the statement is clearly true. $\square$

Proposition 2.2 states the validity of the feasible set inequality for $P(T \cup \{j\})$. To obtain a (strong) valid inequality for P we resort to *lifting*, see Padberg [1975]. Consider some permutation $\pi_1, \dots, \pi_{n-|T|-1}$ of the set $N \setminus (T \cup \{j\})$. For $k = 1, \dots, n - |T| - 1$ and $l = 1, \dots, u_{\pi_k}$ let

$$\begin{array}{lrl} & \gamma(k,l) = & \max \displaystyle\sum_{i \in T \cup \{j\}} w_i x_i + \sum_{i \in \{\pi_1, \dots, \pi_{k-1}\}} w_i x_i \\ (3) & & \displaystyle\sum_{i \in T \cup \{j\}} A_{\cdot i} x_i + \sum_{i \in \{\pi_1, \dots, \pi_{k-1}\}} A_{\cdot i} x_i + A_{\cdot \pi_k} l \leq b \\ & & 0 \leq x_i \leq u_i, x_i \in \mathbb{Z} \text{ for } i \in T \cup \{j, \pi_1, \dots, \pi_{k-1}\}. \end{array}$$

With $\gamma = \sum_{i \in T} w_i u_i$ the lifting coefficients are

$$w_{\pi_k} := \min_{l=1,\dots,u_{\pi_k}} \frac{\gamma - \gamma(k,l)}{l}. \tag{4}$$

The following statement is immediate.

Proposition 2.3 *The (lifted) feasible set inequality $w^T x \leq \sum_{i \in T} w_i u_i$ is valid for P.*

Note that the right-hand side of (2) coincides with (4) applied to variable j if we substitute in (3) the set $T \cup \{j\}$ by T. In other words, a lifted feasible set inequality associated with T and $\{j\}$, where the variables in $N \setminus (T \cup \{j\})$ are lifted according to the sequence $\pi_1, \dots, \pi_{n-|T|-1}$, coincides with the inequality associated with T, where j is lifted first, and the remaining variables $N \setminus (T \cup \{j\})$ are lifted in the same order $\pi_1, \dots, \pi_{n-|T|-1}$. Thus, instead of speaking of a feasible set inequality associated with T and $\{j\}$, we speak in the sequel of a feasible set inequality associated with T and view j as the variable that is lifted first.

Examples of feasible set inequalities include $(1,k)$-configuration, minimal-cover, and extended weight inequalities that are known for the knapsack polytope $P_K(N, a, \alpha) = \text{conv}\{x \in \{0,1\}^N : a^T x \leq \alpha\}$ with $a \in \mathbb{R}^N_+, \alpha > 0$. Let $S \subseteq N$ be a minimal cover, i.e., $a(S) > \alpha$ and $a(S \setminus \{i\}) \leq \alpha$ for all $i \in S$, and partition S into T and $\{j\}$ for some $j \in S$. Set $w_i := 1$ for all $i \in T$. The feasible set inequality reads $\sum_{i \in T} x_i + w_j x_j \leq |T| = |S| - 1$ with $w_j \leq \min\{|V| : V \subseteq T, \sum_{i \in V} a_i \geq a_j - r(T)\}$. Since $\sum_{i \in T} a_i + r(T) = \alpha$ and $\sum_{i \in S} a_i > \alpha$, this minimum is greater than or equal to one. Therefore, under the regularity condition imposed in Padberg [1980], the feasible set inequality is always a $(1,k)$-configuration inequality. In case the coefficient happens to be one we get a minimal cover inequality (Wolsey [1975]). Moreover, feasible set inequalities are just the extended weight inequalities for the knapsack polytope if we choose weights $w_i = 1$ for $i \in T$ (Weismantel [1997]).

Odd hole- and clique inequalities (Padberg [1973], [1975]) for the set packing polytope are further examples of lifted feasible set inequalities. For some 0/1 matrix $A \in \{0,1\}^{M \times N}$, consider the set packing polytope $P(N, M, A, \mathbb{1}, \mathbb{1}) = \text{conv}\{x \in \{0,1\}^N : Ax \leq \mathbb{1}\}$, where $\mathbb{1}$ denotes the all one vector. Let $G_A = (V, E)$ denote the associated column intersection graph whose nodes correspond to the columns of A and nodes i and j are adjacent if and only if the columns associated with i and j intersect in some row. Let $Q \subseteq V$ be a clique in G_A, then the clique inequality $\sum_{i \in Q} x_i \leq 1$ is valid for P. To see that this inequality is a lifted feasible set inequality, let $T = \{i\}$ for some $i \in Q$. The feasible set inequality $x_i \leq 1$ is valid for $P(\{i\})$. Lifting the remaining variables $k \in Q \setminus \{i\}$ by applying formula (4) yields $w_k = 1$, and the clique inequality follows.

3 Some Properties of Feasible Set Inequalities

In this section we summarize some theoretical properties of feasible set inequalities. In particular, we give lower and upper bounds on the exact lifting coefficients and point out a connection to Chvátal-Gomory cutting planes. For a detailed discussion of these issues including proofs we refer to Martin and Weismantel [1998].

Lower an Upper Bounds on the Lifting Coefficients

This is an important issue from a computational point of view, since the calculation of the exact lifting coefficients requires the solution of the integer programs in (3). It is thus desirable to have lower and upper bounds on the lifting coefficient that are easier to compute in certain situations. We assume throughout this section that $A \geq 0$.

Definition 3.1 *Let $T \subseteq N$ be a feasible set and $w : T \mapsto \mathbb{R}^T_+$ a weighting of T. For $v \in \mathbb{R}^m$ we define the*

Covering Number

$$\phi^{\geq}(v) := \min \{\sum_{i \in T} w_i x_i : \sum_{i \in T} A_{\cdot i} x_i \geq v,\, 0 \leq x_i \leq u_i, x_i \in \mathbb{Z}, i \in T\},$$

$\leq$-Incomparability Number

$$\phi^{\nleq}(v) := \min \{\sum_{i \in T} w_i x_i : \sum_{i \in T} A_{\cdot i} x_i \nleq v,\, 0 \leq x_i \leq u_i, x_i \in \mathbb{Z}, i \in T\},$$

where we set $\phi^{\nleq}(v) := 0$ for $v \leq 0$.

Consider a (lifted) feasible set inequality $w^T x \leq \sum_{i \in T} w_i u_i$ associated with T, where the variables in $N \setminus T$ are lifted in the sequence $\pi_1, \dots, \pi_{n-|T|}$.

Theorem 3.2

(a) $w_{\pi_1} = \min_{l=1,\dots,u_{\pi_1}} \frac{1}{l} \phi^{\geq}(A_{\cdot \pi_1} l - r(T))$.

(b) $w_{\pi_k} \leq \min_{l=1,\dots,u_{\pi_k}} \frac{1}{l} \phi^{\geq}(A_{\cdot \pi_k} l - r(T))$, *for* $k = 2, \dots, n - |T|$.

(c) $w_{\pi_k} \geq \min_{l=1,\dots,u_{\pi_k}} \frac{\phi^{\nleq}(A_{\cdot \pi_k} l - r(T))}{l} - \max\{w_i : i \in T\}$, *for* $k = 2, \dots, n - |T|$.

Theorem 3.2 applies, in particular, if we set the coefficient of the first lifted variable w_{π_1} to the upper bound of Theorem 3.2 (a). There are examples where the lower bounds given in Theorem 3.2 are tight.

Connection to Chvátal-Gomory Cuts

In our derivation of feasible set inequalities we have not subsumed any assumptions on the matrix A with the exception that with u_T also every vector $v \leq u_T$ is valid for P. Thus a comparison to Chvátal-Gomory cutting planes that do not rely on any particular structure of A is natural. Recall that Chvátal-Gomory inequalities for the system $Ax \leq b$, $0 \leq x \leq u$, $x \in \mathbb{Z}^n$ are cutting planes $d^T x \leq \delta$ such that $d_i = \lfloor \lambda^T \hat{A}_{\cdot i} \rfloor$, $i = 1, \ldots, n$, and $\delta = \lfloor \lambda^T \hat{b} \rfloor$ for some $\lambda \in \mathbb{R}_+^{m+n}$, where $\hat{A} = \begin{bmatrix} A \\ I \end{bmatrix}$ and $\hat{b} = \begin{bmatrix} b \\ u \end{bmatrix}$.

Consider a (lifted) feasible set inequality $w^T x \leq \sum_{i \in T} w_i u_i$ associated with T, whose remaining variables $N \setminus T$ are lifted in the sequence $\pi_1, \ldots, \pi_{n-|T|}$. This lifted feasible set inequality is compared to Chvátal-Gomory inequalities resulting from multipliers $\lambda \in \mathbb{R}_+^{m+n}$ that satisfy $\lfloor \lambda^T \hat{A}_{\cdot i} \rfloor = w_i$ for $i \in T$.

Proposition 3.3

(a) $\lfloor \lambda^T \hat{b} \rfloor \geq \sum_{i \in T} u_i w_i$.

(b) If $\lfloor \lambda^T \hat{b} \rfloor = \sum_{i \in T} u_i w_i$, *let* j *be the smallest index with* $\lfloor \lambda^T \hat{A}_{\cdot \pi_j} \rfloor \neq w_{\pi_j}$. *Then,* $\lfloor \lambda^T \hat{A}_{\cdot \pi_j} \rfloor < w_{\pi_j}$.

As soon as the first two coefficients differ, for $k \in \{\pi_{j+1}, \ldots, \pi_{n-|T|}\}$, no further statements on the relations of the coefficients are possible, in general. A feasible set- and a Chvátal-Gomory cutting plane sometimes coincide. In these cases it follows that the resulting (lifted) feasible set inequality is independent on the sequence in which the lifting coefficients w_j for the items in $N \setminus T$ are computed.

4 Separating Feasible Set Inequalities

When it comes to incorporate feasible set inequalities into a general mixed integer programming solver many difficulties and questions arise. First, how to find a feasible set T? Consider some general mixed integer program

$$(5) \qquad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \leq b \\ & l \leq x \leq u \\ & x \in \mathbb{Z}^N \times \mathbb{R}^C, \end{array}$$

where N, C and M are finite sets, N and C are disjoint, $A \in \mathbb{R}^{M \times (N \cup C)}$, and l, u, c and b are vectors of appropriate dimension. In general, the given constraint matrix A contains positive and negative entries, and it is a-priori not clear, how to guarantee, for some subset $T \subseteq N$, that every vector $v \leq u_T$ is feasible for (5). Second, once one determined a feasible set T, what is a suitable weighting w for T as proposed in Definition 2.1? In principle, every vector non-negative $w \in \mathbb{Z}_+^T$ is possible. Third, how to perform the lifting? The calculation of the exact lifting coefficient for some variable x_j requires the solution of u_j many integer programs, which might be very expensive to compute. Fourth, what are good substructures of A to start with and look for feasible set inequalities. It is pointless to begin with the whole constraint matrix A, since in the same time we find one cutting plane we probably would solve the entire problem. Desirable would be to have the matrix A in so-called bordered block diagonal form, where the matrix consists of a set of small independent blocks that are possibly connected by some linking constraints (the border). For such a structure the blocks are independent of each other and, assuming they are small, we might even perform the lifting exactly. Questions over questions.

It goes without saying that most of these questions can only be answered by computational experiences. We discuss some of the options below. Some properties of feasible set inequalities, however, leave us no choice and restrict our search for good substructures of A. Let us call a substructure of A a *block*. One such property of a feasible set inequality is that it is defined for integer variables with finite lower and upper bounds only. Thus, we remove from our considerations all rows that contain continuous variables or variables with infinite lower or upper bound. Furthermore, we can restrict ourselves to fractional variables. If we are faced with some block containing only variables whose current LP values are integer, there cannot exist a violated feasible set inequality. Thus, in our choice of the block we only take fractional variables into account. The remaining, currently integer variables, will be lifted afterwards. This fact has an important advantage, since the number of fractional variables is usually only some fraction of the total number of variables (often below 10%), and restricts our search for possible blocks to a small part of A.

In detail, we proceed as follows in order to find a reasonable substructure $Bx \leq d$ of $Ax \leq b$, i.e., subsets $I \subseteq M$, $J \subseteq N$ with $B = A_{I,J}$ and $d = b_J$. As outlined possible candidates for I are rows that solely contain integer variables with finite lower and upper bounds and that contain at least one variable whose current LP value is fractional. Second, we restrict ourselves to rows whose number of non-zeros lies between three and some upper bound σ. 'Three' is not really a restriction, since if some row has only two non-zero entries a complete description of the associated integer knapsack polytope is readily available. The number σ is a parameter, which basically is used to control the size of the final block. We have experimented with this parameter a lot, and a reasonable setting on our test set is $\sigma = 20$. We also exclude rows from further consideration that are not tight for the current LP solution. Let us denote by $I_c \subseteq M$ the set of candidate rows. We now construct the so-called row intersection graph, i.e., the graph that is obtained by introducing a node for each row in I_c and an edge whenever two rows intersect in some fractional column. It turns out that due to the fact that usually the number of fractionals is small the row intersection graph for most problems decomposes into components of small size. 'Small' here means no more than three/four rows and up to thirty/forty variables per component. For only very few instances the row intersection graph contains big components. In the latter case, we decompose the big components heuristically. Motivated by the success of the greedy-type algorithms in Borndörfer, Ferreira, and Martin [1997] we start with a row of a component that has largest degree and iteratively add further rows in order of decreasing intersection with the already selected rows. Our current setting allows at most ten rows or around fifty variables in a block. If one of the limits is reached we stop.

After applying this procedure we obtain index sets I_1 and J_1 that define a substructure $Bx \leq d$ of $Ax \leq b$ that contains at most ten rows and up to fifty variables. We delete the rows from I_c whose support intersects with J_1. With the remaining rows we apply the same procedure as described, and we end up with mutually disjoint sets $I_1, \ldots, I_\beta \subseteq M$ and mutually disjoint sets $J_1, \ldots, J_\beta \subseteq N$. For each block $A_{I_b,J_b} x_{J_b} \leq b_{I_b}$ we call our separation algorithm to derive violated feasible set inequalities. The following algorithm outlines our procedure and provides our answers to the questions raised at the beginning of the section. An explanation of the steps will be given afterwards.

Algorithm 4.1 *Separation algorithm for feasible set inequalities.*

Input: *A block* $A_{I,J}x_J \leq b_I$, $l_J \leq x_J \leq u_J$ *of* $Ax \leq b$, $l \leq x \leq u$ *with* $I \subseteq M$ *and* $J \subseteq N$*; an optimal solution* $\bar{x}$ *of the current LP.*

1. *Fix all variables that are integer.*

$$f_i = \begin{cases} \bar{x}_i, & \text{if } \bar{x}_i \in \mathbb{Z}; \\ FREE, & \text{if } i \in \mathrm{Fr}(\bar{x}) := \{j \in J : \bar{x}_j \notin \mathbb{Z}\}. \end{cases}$$

2. *Solve*

$$(6) \qquad \begin{array}{ll} \max & \sum_{i \in \mathrm{Fr}(\bar{x})} \frac{\bar{x}_i}{u_i} z_i \\ & \sum_{i \in \mathrm{Fr}(\bar{x})} A_{\cdot i} z_i \leq b - \sum_{i \notin \mathrm{Fr}(\bar{x})} A_{\cdot i} f_i \\ & 0 \leq z \leq u_{\mathrm{Fr}(\bar{x})},\ z \in \mathbb{Z}^{\mathrm{Fr}(\bar{x})} \end{array}$$

3. **If** *(6) is infeasible, try to derive a violated* bound inequality*.*
   Else *let* y *be an optimal solution of (6).*
   Set $T := \mathrm{supp}(y)$*.*
4. *Assign weights* w_j *for* $j \in T$ *according to*

$$w_j = \begin{cases} +1, & \text{if } |\{i \in I : a_{ij} > 0\}| \geq |\{i \in I : a_{ij} < 0\}|; \\ -1, & \text{else,} \end{cases}$$

   and set $f_i := l_i$ *for* $i \in \mathrm{Fr}(\bar{x}) \setminus T$*.*
5. *Solve*

$$(7) \qquad \begin{array}{ll} \omega = \max & \sum_{i \in T} w_i z_i \\ & \sum_{i \in T} A_{\cdot i} z_i \leq b - \sum_{i \in J \setminus T} A_{\cdot i} f_i \\ & 0 \leq z \leq u_T,\ z \in \mathbb{Z}^T. \end{array}$$

6. $\sum\limits_{i \in T} w_i x_i \leq \omega$ *is the basic feasible set inequality.*
7. *Determine a lifting sequence* $\pi_1, \ldots, \pi_{|J|-|T|}$ *of* $J \setminus T$ *such that all fractional variables come first, i. e.,* $\{\pi_1, \ldots, \pi_{|\mathrm{Fr}(\bar{x}) \setminus T|}\} = \mathrm{Fr}(\bar{x}) \setminus T$*.*
8. **For** $k = 1, \ldots, |J| - |T|$ *perform the following steps.*
   Determine lifting coefficient w_{π_k} *for variable* x_{π_k}*.*
   If $k > |\mathrm{Fr}(\bar{x}) \setminus T|$ *and the current feasible set inequality is not violated –* **Stop** *(no violated inequality found).*
9. *Check whether the* lifted feasible set inequality $w^T x \leq \omega$ *is violated.*
10. **Stop.**

Steps 2, 3 and 5 of Algorithm 4.1 show our choice of determining a "feasible" set T. The integer program (6) is solved to optimality using our general mixed integer programming solver, called SIP. A description of this algorithm can be found in Martin [1998]. Note that as outlined above the number of fractional variables is usually very small resulting in integer programs of small size. Thus, the expected time to solve this integer program is small, and our computational results in the next section confirm this. For the integer program (7) we supply a limit on the number of branch-and-bound nodes of 1000. If we are not able to solve the problem within this limit we use $\omega = \lfloor \hat{\omega} \rfloor$, where $\hat{\omega}$ is the best upper bound after 1000 branch-and-bound nodes. We experimented with several different objective functions in (6), the one here performed best on average. Note that the set T as defined in Step 3 need not be "feasible" as required in Definition 2.1, since it is not guaranteed that every vector $v \leq u_T$ is feasible. This strong requirement is hardly satisfied in practice. To guarantee validity of the basic feasible set inequality a much weaker condition suffices, namely, every feasible solution v of (7) must satisfy $\sum_{i \in T} w_i v_i \leq \sum_{i \in T} w_i u_i$. This is one reason for solving (7), which determines the maximal possible right-hand side. Note that this value ω might be greater or less than $\sum_{i\in T} w_i u_i$, since $w_i = \pm 1$ for $i \in T$. However, if $w \geq 0$ the number ω is often much smaller than $\sum_{i\in T} w_i u_i$, especially if some integer variables have large upper bounds. Although the set T might not satisfy the requirements of Definition 2.1, we call it still *feasible* in the following. Our weighting w is restricted to take ± 1 values only. We also tried different weightings with limited success. Note also that if (6) does not have a feasible solution, see Step 3, and if all currently integer variables are fixed at their lower or upper bounds, i.e., $f_i \in \{l_i, u_i\}$ for all $i \notin \mathrm{Fr}(\bar{x})$, the inequality

$$\sum_{\{i: f_i = l_i\}} (x_i - l_i) + \sum_{\{i: f_i = u_i\}} (u_i - x_i) \geq 1, \tag{8}$$

called *bound inequality*, is valid for $P = \mathrm{conv}\{x \in \mathbb{R}^{N\cup C} : x \text{ feasible for } (5)\}$. This inequality is also violated by the current LP solution $\bar{x}$ and we return with this violated inequality from Algorithm 4.1.

It remains to discuss the lifting, Steps 7 and 8. The lifting sequence starts with all remaining fractional variables, sorted in non-increasing order of their LP value, followed by all integer variables that are not at one of their bounds, i.e., $l_i < f_i < u_i$, and last, we lift all integer variables that are currently at one of their bounds. For the elements in these two latter sets we do not impose any particular order. The lifting of one variable basically follows the standard scheme (Padberg [1975]) with some adjustments. First, we have to take into account that certain variables are fixed to some non-zero value. This can easily be achieved as outlined below. A more serious problem is the lifting of variables that are currently neither fixed on their lower nor on their upper bound. In this case, it might be possible that the lifted inequality is no longer valid. Before discussing this difficulty in detail, let us describe the overall algorithm.

Algorithm 4.2 *Lifting variable x_{π_k}.*

Input: *An inequality $\sum_{i\in S} w_i x_i \leq \omega$ valid for $P\,(I, S, A_{I,S}, b - \sum_{i\in N\setminus S} A_{\cdot i} f_i, u_S)$, where $S := T \cup \{\pi_1, \ldots, \pi_{k-1}\}$.*

Output: *Lifting coefficient w_{π_k} for x_{π_k} and (new) right-hand side ω such that*

$$\sum_{i\in S\cup\{\pi_k\}} w_i x_i \leq \omega \tag{9}$$

is valid for $P\,(I, S\cup\{\pi_k\}, A_{I,S\cup\{\pi_k\}}, b - \sum_{i\in N\setminus(S\cup\{\pi_k\})} A_{\cdot i} f_i, u_{S\cup\{\pi_k\}})$.

1. **For** $l = l_i, \ldots, u_i$ *compute*

$$\begin{aligned} \omega_l = \max\quad & \textstyle\sum_{i\in S} w_i z_i \\ & \textstyle\sum_{i\in S} A_{\cdot i} z_i + A_{\cdot \pi_k} \cdot l \leq b - \sum_{i\in N\setminus S} A_{\cdot i} f_i \\ & 0 \leq z \leq u_S,\ z \in \mathbb{Z}^S. \end{aligned} \tag{10}$$

2. Let

$$\begin{aligned} w^- &= \max_{l = l_{\pi_k}, \ldots, f_{\pi_k}-1} \frac{\omega - \omega_l}{l - f_{\pi_k}} \\ w^+ &= \min_{l = f_{\pi_k}+1, \ldots, u_{\pi_k}} \frac{\omega - \omega_l}{l - f_{\pi_k}} \end{aligned} \tag{11}$$

3. **If** $f_{\pi_k} = l_{\pi_k}$ *set*

$$\begin{aligned} w_{\pi_k} &= w^+ \\ \omega &= \omega + w^+ f_{\pi_k} \end{aligned} \tag{12}$$

4. **If** $f_{\pi_k} = u_{\pi_k}$ *set*

$$(13) \qquad \begin{aligned} w_{\pi_k} &= w^- \\ \omega &= \omega + w^- f_{\pi_k} \end{aligned}$$

5. **If** $l_{\pi_k} < f_{\pi_k} < u_{\pi_k}$ *perform the following steps:*
If $w^- \leq w^+$ *return with (12) or (13).*
If $w^- > w^+$ *return with*

$$(14) \qquad \begin{aligned} w_{\pi_k} &= w^- \\ \omega &= \omega + w^- f_{\pi_k} + (u_{\pi_k} - f_{\pi_k})(w^- - w^+) \end{aligned}$$

or with

$$(15) \qquad \begin{aligned} w_{\pi_k} &= w^+ \\ \omega &= \omega + w^- f_{\pi_k} - l_{\pi_k}(w^- - w^+) \end{aligned}$$

or with

$$(16) \qquad \begin{aligned} w_{\pi_k} &= \frac{u_{\pi_k} - l_{\pi_k} - f_{\pi_k}}{u_{\pi_k} - l_{\pi_k}}(w^+ - w^-) + w^- \\ \omega &= \omega + w^- f_{\pi_k} - \frac{u_{\pi_k} - l_{\pi_k} - f_{\pi_k}}{u_{\pi_k} - l_{\pi_k}}(w^+ - w^-) l_{\pi_k}\,, \end{aligned}$$

the latter is only valid if $l_{\pi_k} + f_{\pi_k} \leq u_{\pi_k}$.

6. **Stop.**

The following proposition shows that the settings in Steps 3 and 4 of Algorithm 4.2 are correct.

Proposition 4.3

(a) If $f_{\pi_k} \in \{l_{\pi_k}, u_{\pi_k}\}$*, the settings in (12) and (13) yield a valid inequality.*

(b) If $l_{\pi_k} < f_{\pi_k} < u_{\pi_k}$ *and* $w^- \leq w^+$*, both (12) and (13) yield a valid inequality.*

Proof. Let $\bar{x} \in P\left(I, S \cup \{\pi_k\}, A_{I,S\cup\{\pi_k\}}, b - \sum_{i\in N\setminus(S\cup\{\pi_k\})} A_{\cdot i} f_i, u_{S\cup\{\pi_k\}}\right)$ be integer.

Consider part (a) and the case $f_{\pi_k} = l_{\pi_k}$ and denote by $\bar{\omega} = \omega + w^+ f_{\pi_k}$ the new right-hand side. Note that the lifted inequality (9) can be rewritten as

$$\sum_{i\in S} w_i x_i + w_{\pi_k}(x_{\pi_k} - l_{\pi_k}) \leq \omega$$

Thus, if $\bar{x}_{\pi_k} = l_{\pi_k}$, the inequality is trivially valid. If $\bar{x}_{\pi_k} = l$ for some $l_{\pi_k} < l < u_{\pi_k}$, we obtain $\sum_{i\in S} w_i \bar{x}_i + w_{\pi_k}(\bar{x}_{\pi_k} - l_{\pi_k}) \leq \sum_{i\in S} w_i \bar{x}_i + \frac{\omega - \omega_l}{l - l_{\pi_k}}(l - l_{\pi_k}) \leq \omega_l + (\omega - \omega_l) = \omega$. The same proof applies if $f_{\pi_k} = u_{\pi_k}$.

We show (b) for the settings in (12), i. e., we prove validity of $\sum_{i\in S} w_i x_i + w^+(x_{\pi_k} - f_{\pi_k}) \leq \omega$. (13) can be shown analogously. If $\bar{x}_{\pi_k} \in \{f_{\pi_k}, \ldots, u_{\pi_k}\}$ the validity follows by part (a). If $\bar{x}_{\pi_k} \in \{l_{\pi_k}, \ldots, f_{\pi_k} - 1\}$, we get $\sum_{i\in S} w_i \bar{x}_i + w^+(\bar{x}_{\pi_k} - f_{\pi_k}) \leq \sum_{i\in S} w_i x_i + w^-(x_{\pi_k} - f_{\pi_k}) \leq \omega$. The first inequality follows from $\bar{x}_{\pi_k} - f_{\pi_k} < 0$ and $w^- > w^+$, the second from part (a). □

It remains to discuss the case $l_{\pi_k} < f_{\pi_k} < u_{\pi_k}$ and $w^- > w^+$. In this case none of the two settings (12) and (13) needs to be valid as the following example shows.

Example 4.4 *Consider the integer program*

$$\begin{aligned} \min \quad & 4200\,x_1 + 1500\,x_2 + 2700\,x_3 \\ & 0.9\,x_1 + x_2 - x_3 = 5.7 \\ & 0 \leq x_1 \leq 15 \\ & 0 \leq x_2 \leq 18 \\ & 0 \leq x_3 \leq 18,\ x \in \mathbb{Z}^3. \end{aligned}$$

This is a substructure of example flugpl *from the* `Miplib`*(see the description in Section 5). Running* `SIP` *for this problem yields an optimal solution for an LP encountered during the solution process, where the three variables have values $x_1 = 5.5, x_2 = 3.75$, and $x_3 = 3$. Thus we fix variable x_3 to three, i. e., $f_3 = 3$. The inequality $x_1 + x_2 \le 9$ is valid for* $\mathrm{conv}\{x \in \mathbb{Z}^2 : 0.9x_1 + x_2 = 8.7, 0 \le x_1 \le 15, 0 \le x_2 \le 18\}$. *Now in order to lift x_3 we compute w^+ and w^- in (11). We get $w^+ = -\frac{4}{3}$ and the associated inequality (12) reads $x_1 + x_2 - \frac{4}{3}x_3 \le 5$. It is violated by the feasible solution $x = (3, 3, 0)^T$. Similarly, $w^- = -1$ and the inequality $x_1 + x_2 - x_3 \le 6$ from (13) is violated by the feasible solution $x = (13, 0, 6)^T$.*

The problem discussed in Example 4.4 is not new. Balas, Ceria, Cornuéjols, and Natraj [1996] give another example in conjunction with Chvátal-Gomory cuts. Here the same difficulty arises, when Chvátal-Gomory cuts are generated at some node, which is not the root note. When integer variables are involved these inequalities are usually only locally valid for this node and its descendants, but not for the entire branch-and-bound tree. Balas, Ceria, Cornuéjols, and Natraj [1996] show that cuts are indeed globally valid if all integer variables are binary.

Algorithm 4.2 gives one possible answer to the general integer case. The idea is to consider two polytopes P^+ and P^-. P^+ is the original polytope intersected with the inequality using formula (12) and the bound constraints $f_{\pi_k} \le x_{\pi_k} \le u_{\pi_k}$. P^- is defined accordingly, i. e.,

$$\begin{array}{ll} (17) \qquad & P^- = \{x \in P : \sum_{i \in S} w_i x_i + w^- x_{\pi_k} \le \omega + w^- f_{\pi_k}, l_{\pi_k} \le x_{\pi_k} \le f_{\pi_k}\}. \\ & P^+ = \{x \in P : \sum_{i \in S} w_i x_i + w^+ x_{\pi_k} \le \omega + w^+ f_{\pi_k}, f_{\pi_k} \le x_{\pi_k} \le u_{\pi_k}\}. \end{array}$$

Now we can apply the idea of disjunctive programming and determine $\mathrm{conv}(P^- \cup P^+)$ to derive a valid inequality for $P\,(S \cup \{\pi_k\})$.

Let us briefly summarize the concept of disjunctive programming applied to our case, for more details see Balas, Ceria, and Cornuéjols [1993] and the references therein. Suppose for the ease of exposition that $P^+ = \{x \in \mathbb{R}^N : A^1 x \le b^1\}$ and $P^- = \{x \in \mathbb{R}^N : A^2 x \le b^2\}$. Now

$$(18) \qquad \begin{array}{rl} \mathrm{conv}(P^- \cup P^+) = \{x \in \mathbb{R}^N : & \text{there exist } \bar{y}^1, \bar{y}^2, \lambda^1, \lambda^2 \text{ such that:} \\ & (a)\ \ x = \lambda^1 \bar{y}^1 + \lambda^2 \bar{y}^2 \\ & (b)\ \ A^1 \bar{y}^1 \le b^1 \\ & (c)\ \ A^2 \bar{y}^2 \le b^2 \\ & (d)\ \ \lambda^1 + \lambda^2 = 1, \lambda^1 \ge 0, \lambda^2 \ge 0\}. \end{array}$$

(18) (a) contains quadratic terms. Multiplying (18) with λ_i and setting $y^i := \lambda_i \bar{y}^i$ for $i = 1, 2$ yields the polyhedron

$$(19) \qquad \begin{array}{rl} Q = \{x \in \mathbb{R}^N : & \text{there exist } y^1, y^2, \lambda^1, \lambda^2 \text{ such that:} \\ & (a)\ \ x = y^1 + y^2 \\ & (b)\ \ A^1 y^1 \le \lambda^1 b^1 \\ & (c)\ \ A^2 y^2 \le \lambda^2 b^2 \\ & (d)\ \ \lambda^1 + \lambda^2 = 1, \lambda^1 \ge 0, \lambda^2 \ge 0\}. \end{array}$$

Q still coincides with $\mathrm{conv}(P^- \cup P^+)$ if we guarantee that $y^i = 0$ whenever $\lambda_i = 0$. This is true, since P^+ and P^- are bounded implying that $\{y^i \in \mathbb{R}^n : Ay^i \le 0\} = \{0\}$ for $i = 1, 2$.

Let us abbreviate the set of vectors $x, y^1, y^2, \lambda^1, \lambda^2$ satisfying (19) (a) – (d) by $L = \{(x, z)^T : Dx + Bz \le d\}$, where $z = (y^1, y^2, \lambda^1, \lambda^2)^T$ and B, D, d are appropriate matrices and vectors. Q, and $\mathrm{conv}(P^- \cup P^+)$, is the projection of L onto the x-space, i. e.,

$$\mathrm{conv}(P^- \cup P^+) = \{x \in \mathbb{R}^N : \text{there exists } z \text{ such that } (x, z)^T \in L\}.$$

In order to obtain from L a description of $\operatorname{conv}(P^- \cup P^+)$ by means of linear inequalities, we need to eliminate the z-variables. With

$$C = \{v : \; v^T B = 0, \, v \geq 0\} \tag{20}$$

we get

$$\operatorname{conv}(P^- \cup P^+) = \{x \in \mathbb{R}^N \, : \, (v^T D)x \leq v^T d \text{ for all } v \in C\}.$$

In order to find a valid inequality that cuts off the LP solution $\bar{x}$ we solve the linear program

$$\begin{array}{ll} \max & (D\bar{x} - d)^T v \\ & v \in C. \end{array} \tag{21}$$

Note that if there is a violated inequality in C, then (21) is unbounded, since C is a polyhedral cone. For algorithmic conveniences C is often truncated by bounding the vector v with respect to some norm. Solving the linear program in (21) yields an inequality that is valid for P. If the optimum in (21) is zero, there exists no violated inequality, otherwise $(\bar{v}^T D)x \leq \bar{v}^T d$ yields the desired inequality, where $\bar{v}$ is an optimal solution (extreme ray) of (21).

We apply this scheme in our implementation to the following two polytopes

$$\begin{array}{l} \tilde{P}^- = \{x \in \mathbb{R}^N : \sum_{i \in S} w_i x_i + w^- x_{\pi_k} \leq \omega + w^- f_{\pi_k}, l_{\pi_k} \leq x_{\pi_k} \leq f_{\pi_k}\}, \\ \tilde{P}^+ = \{x \in \mathbb{R}^N : \sum_{i \in S} w_i x_i + w^+ x_{\pi_k} \leq \omega + w^+ f_{\pi_k}, f_{\pi_k} \leq x_{\pi_k} \leq u_{\pi_k}\}. \end{array} \tag{22}$$

That is, we consider a relaxation of the polytopes defined in (17) by neglecting the constraints defining P. The advantage of using this relaxation is that – as we will see in a moment – we can solve the resulting linear program (21) explicitly. The drawback, of course, is that we might get a weaker inequality. The linear program (21) that results when using $\tilde{P}^-$ and $\tilde{P}^+$ in (22) has (after some obvious variable substitutions) seven variables and two constraints. We fix the variable v_i to one that corresponds to the constraint $\sum_{i \in S} w_i x_i + w^- x_{\pi_k} \leq \omega + w^- f_{\pi_k}$. This way, we guarantee that the coefficients w_i for $i \in S$ stay unchanged in the resulting lifted inequality. After doing some calculations it turns out this linear program, with now six variables and two constraints, has only three possible optimal solutions. The three LP solutions yield three different inequalities as stated in (14), (15), and (16). We use the one that maximizes the slack.

5 Computational Results

In this section we report on our computational experience with Algorithm 4.2 for the feasible set inequalities. We have incorporated this algorithm in `SIP`, and tested it on the instances of the mixed integer programming library, `Miplib` (see Bixby, Ceria, McZeal, and Savelsbergh [1998]). We compared `SIP` with the default parameter setting and `SIP` where in addition the separation algorithm for feasible set inequalities has been turned on. We use a time limit of 3600 CPU seconds and limit the number of branch-and-bound nodes to one million. The tests were performed on a Sun Enterprise 3000 with a 168 MHz UltraSPARC processor and 1024 MB main memory.

It turns out that for 48 out of 59 `Miplib`-problems we do not find feasible set inequalities and the overhead for applying our separation algorithm is below 1% of the total running time. For the remaining 11 problems we find feasible set inequalities or our separation routine uses more than 1% of the computation time. Tables 1 and 2 show the results for these examples. Column 1 gives the problem name followed by the number of branch-and-bound nodes in Column 2.

The next two columns give the number of cuts found, *Others* include all cuts that are separated by default in `SIP`. A significant amount of these cuts are 0/1-knapsack inequalities. Column *FS* give the number of feasible set inequalities found. Columns 5 and 6 show timings, the time used in Algorithm 4.2 and the total running time. If we cannot solve the problem within the time limit of 3600 seconds, the last Column *Gap %* shows a non-zero gap ($= 100\, \frac{|\text{upper bound - lower bound}|}{|\text{lower bound}|}$).

Example	B & B	Cuts		Time		Gap %
		Others	FS	FS	Total	
fiber	783	372	0	0.0	16.9	0.000
gesa2	209525	33	0	0.0	3600.0	0.048
gesa3_o	74472	0	0	0.0	1144.7	0.000
misc03	699	14	0	0.0	4.1	0.000
misc07	35585	0	0	0.0	378.8	0.000
p0033	57	32	0	0.0	0.1	0.000
p0201	507	136	0	0.0	5.0	0.000
p2756	23151	6923	0	0.0	3600.2	0.891
seymour	1947	0	0	0.0	3601.8	7.770
stein27	4666	0	0	0.0	8.0	0.000
stein45	54077	0	0	0.0	277.7	0.000
Total (11)	405469	7510	0	0.0	12637.3	8.709

Table 1: `SIP` without feasible set inequalities

Example	B & B	Cuts		Time		Gap %
		Others	FS	FS	Total	
fiber	771	358	1	3.3	19.4	0.000
gesa2	211168	33	4	9.0	3600.0	0.047
gesa3_o	74472	0	0	70.7	1218.4	0.000
misc03	703	14	11	1.1	5.4	0.000
misc07	20962	0	365	54.1	333.0	0.000
p0033	75	40	5	0.6	0.7	0.000
p0201	507	136	1	2.0	7.1	0.000
p2756	23041	6842	6	4.9	3600.9	0.803
seymour	2146	0	10	5.4	3600.0	5.726
stein27	3660	0	142	1.6	10.0	0.000
stein45	47612	0	783	23.6	308.5	0.000
Total (11)	385117	7423	1328	176.4	12703.4	6.576

Table 2: `SIP` with feasible set inequalities

Comparing Tables 1 and 2 we see that we can significantly reduce the gap for test problem seymour and slightly for gesa2 and p2756, resulting in a total reduction of about 25%. We also recognize that basically no time is spent for separating feasible set inequalities. This is very astonishing, since almost all integer programs that come up in Algorithm 4.2 (and these are thousands) are solved to optimality, where we use an upper bound on the branch-and-bound nodes of 1000. The success of the feasible set inequalities relies on the fact that we *can* solve small integer programs to optimality in general very fast. There are other interesting facts that can be read from Table 2. The "stein"-problems, where we find a considerable amount of cuts, and example seymour, where we improve the quality substantially, are set covering problems. There seem to be virtually no efficient separation algorithms for set covering problems. To the best of our knowledge the only exceptions are the cutting planes from conditional bounds by Balas and Ho [1980], a class of k-projection inequalities by Nobili and Sassano [1992], and aggregated cycle inequalities by Borndörfer and Weismantel [1997]. It is therefore even more astonishing that our general separation algorithm for feasible set inequalities finds violated inequalities for this type of problems.

The only example where we spent a significant amount of time in Algorithm 4.2 and fail to find any violated cut is example gesa3_o. We looked at this example in detail and it turned out that the inequalities of the identified blocks already give a complete description of the polytope induced by this block. These means, that the fractional solution must be a convex combination of feasible integer solutions and we fail to find violated cuts. The question is, of course, how to avoid this case. One way to overcome this difficulty is to first check, whether the current fractional solution is a convex combination of feasible integer solutions, and then to start the search for violated cuts. This approach has been successfully used by Applegate, Bixby, Chvátal, and Cook [1998] for the solution of traveling salesman problems and there is good hope that their ideas might be carried over to our case, i. e., to general integer programs.

References

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (1998). Project-and-lift (a paradigm for finding cuts). Draft.

Balas, E., Ceria, S., and Cornuéjols, G. (1993). A lift-and-project cutting plane algorithm for mixed $0-1$ programs. *Mathematical Programming*, 58:295–324.

Balas, E., Ceria, S., Cornuéjols, G., and Natraj, N. (1996). Gomory cuts revisited. *Operations Research Letters*, 19:1 – 9.

Balas, E. and Ho, A. (1980). Set covering algorithms using cutting planes, heuristics, and subgradient optimization: A computational study. *Mathematical Programming*, 12:37–60.

Bixby, R., Ceria, S., McZeal, C., and Savelsbergh, M. (1998). An updated mixed integer programming library: Miplib 3.0. Paper and Problems available at WWW Page: http://www.caam.rice.edu/~bixby/miplib/miplib.html.

Borndörfer, R., Ferreira, C., and Martin, A. (1997). Decomposing matrices into blocks. Preprint SC 97-15, Konrad-Zuse-Zentrum Berlin. To appear in SIAM Journal on Optimization.

Borndörfer, R. and Weismantel, R. (1997). Relations among some combinatorial programs. Technical Report Preprint SC 97-54, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

Ceria, S., Cordier, C., Marchand, H., and Wolsey, L. (1998). Cutting planes for integer programs with general integer variables. *Mathematical Programming*, 81:201 – 214.

Crowder, H., Johnson, E., and Padberg, M. (1983). Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834.

Marchand, H. and Wolsey, L. (1997). The $0-1$ knapsack problem with a single continuous variable. Technical Report CORE DP9720, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

Martin, A. (1998). Integer programs with block structure. Habilitations-Schrift, Technische Universität Berlin.

Martin, A. and Weismantel, R. (1998). The intersection of knapsack polyhedra and extensions. In Bixby, R., Boyd, E., and Ríos-Mercado, R., editors, *Integer Programming and Combinatorial Optimization*, Proceedings of the 6th IPCO Conference, pages 243 – 256.

Nobili, P. and Sassano, A. (1992). A separation routine for the set covering polytope. In Balas, E., Cornuéjols, G., and Kannan, R., editors, *Integer Programming and Combinatorial Optimization*, Proceedings of the 2nd IPCO Conference, pages 201 – 219.

Padberg, M. (1973). On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215.

Padberg, M. (1975). A note on zero-one programming. *Operations Research*, 23:833–837.

Padberg, M. (1980). $(1, k)$-configurations and facets for packing problems. *Mathematical Programming*, 18:94–99.

Padberg, M., Roy, T. V., and Wolsey, L. (1985). Valid inequalities for fixed charge problems. *Operations Research*, 33:842 – 861.

Roy, T. V. and Wolsey, L. (1987). Solving mixed integer programming problems using automatic reformulation. *Operations Research*, 35:45 – 57.

Weismantel, R. (1997). On the 0/1 knapsack polytope. *Mathematical Programming*, 77:49–68.

Wolsey, L. (1975). Faces of linear inequalities in 0-1 variables. *Mathematical Programming*, 8:165 – 178.