Discrete Applied Mathematics 123 (2002) 397–446

DISCRETE APPLIED MATHEMATICS

Cutting planes in integer and mixed integer programming ☆

Hugues Marchand[a,1], Alexander Martin[b], Robert Weismantel[c,2], Laurence Wolsey[d,*]

[a] *Electrabel, Boulevard du Regent 8, 1000 Brussels, Belgium*
[b] *Fachbereich Mathematik, Technische Universität Darmstadt, Schlossgartenstr, 7, 64289 Darmstadt, Germany*
[c] *Fakultät für Mathematik, IMO Otto-von-Guericke Universität Magdeburg Universitätsplatz 2 D-39106 Magdeburg, Germany*
[d] *CORE and INMA, Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium*

Received 9 September 1999; received in revised form 30 March 2001; accepted 2 April 2001

Abstract

This survey presents cutting planes that are useful or potentially useful in solving mixed integer programs. Valid inequalities for (i) general integer programs, (ii) problems with local structure such as knapsack constraints, and (iii) problems with 0–1 coefficient matrices, such as set packing, are examined in turn. Finally, the use of valid inequalities for classes of problems with structure, such as network design, is explored. © 2002 Elsevier Science B.V. All rights reserved.

MSC: 90C10; 90C11

Keywords: Mixed integer programming; Cutting planes

☆ Research carried out with financial support of the project TMR-DONET nr. ERB FMRX-CT98-0202 of the European Community.

* Corresponding author. Tel.: +32-10-47-4307; fax: +32-10-47-4301.
E-mail address: wolsey@core.ucl.ac.be (L. Wolsey).
[1] Work carried out at the Department of Operational Research London School of Economics, Houghton Street, London WC2A 2AE, United Kingdom.

[2] Supported by a "Gerhard-Hess-Forschungsförderpreis" (WE 1462/2-1) of the German Science Foundation (DFG).

PII: S0166-218X(01)00348-1

0. Introduction

This survey is devoted to cutting planes that are useful or potentially useful in solving mixed integer programs. This topic is important because (a) improving formulations with cutting planes is of interest independently of the algorithm used to solve the problem, and (b) linear programming based branch-and-bound with cuts added, known as *branch-and-cut*, is now one of the most widespread and successful tools for solving mixed integer programs.

The paper is divided into four sections. First, we discuss ways of generating cuts for general integer programs (IPs) $\max\{c^{\mathrm{T}}x\colon Ax=b,\ x\in\mathbb{Z}_+^n\}$ and mixed integer programs (MIPs) $\max\{c^{\mathrm{T}}x+h^{\mathrm{T}}y\colon Ax+Gy=b, x\in\mathbb{Z}_+^n, y\in\mathbb{R}_+^p\}$ independently of any problem structure. It was shown theoretically in the 70s and 80s that Gomory's mixed integer cuts, simple disjunctive cuts and mixed integer rounding cuts are based on the same disjunctive argument. In the 90s it has been shown, starting with lift-and-project (disjunctive) cuts, how all three types of cuts can be successfully used computationally.

In Section 2.1 we look at IPs and MIPs with some local structure, starting with knapsack sets. Whereas cover inequalities for 0–1 knapsack sets were studied and used in the 70s and 80s, attention has switched to various generalizations of 0–1 knapsack sets, in particular several mixed knapsack sets containing one or more continuous variables. These have a richer polyhedral structure then the pure knapsack sets, and arise very naturally in mixed integer programs. We introduce lifting, an important technique for strengthening valid inequalities and obtaining facet-defining inequalities. Still in the context of knapsack sets, we also introduce a new way to derive valid inequalities, starting from feasible solutions. Knapsack constraints arise when studying IPs whose constraint matrices have general integer coefficients. We also consider problems with 0–1 coefficient matrices. A natural starting point is the set packing problem. The basic inequalities for the set packing polytope based on cliques and cycles are derived, as well as the separation problem for such inequalities. We then briefly examine the generalizations of these inequalities to general independence systems.

A major challenge is to produce stronger inequalities in a way that is easily characterized, and potentially useful for computation. In Section 3 some steps in this direction are examined. A procedure to mix mixed integer rounding inequalities is presented, and also a way to extend formulations of certain combinatorial optimization problems to include set packing relaxations. Though their computational significance is still to be demonstrated, we discuss polynomial algorithms relating to the Lovász-Schrijver lift-and-project procedure, semi-definite optimization and clique separation.

In Section 4 we look at four important problem classes, ranging from network design to electricity generation, and try to indicate the state-of-the-art in terms of known strong cutting planes, and their use in computation.

We assume that readers are familiar with elementary terminology of valid inequalities and polyhedra, see for instance [78,95,107] for an in-depth treatment. See also [62] for a recent discussion of computational issues.

1. General cutting planes

In this section, we discuss methods of generating cutting planes for general mixed integer programs without exploiting any problem structure. As we will see, in certain cases these methods provide a complete linear description of the polyhedron under consideration. As a warm-up we start with the pure integer case and describe the well known Chvátal–Gomory cutting planes. We will see that this approach (based on a rounding argument) fails if continuous variables are involved. Methods that apply to the general mixed integer case are based on a disjunctive argument, and we will discuss three of them.

1.1. Pure integer programs

Consider a pure integer program $\min\{c^{\mathrm{T}}x:\ x \in X\}$ where $X = \{x \in \mathbb{Z}^n_+:\ Ax = b\}$ and A, b are integer. Gomory and later Chvátal found distinct but closely related ways of finding a linear description of $\mathrm{conv}(X)$. We begin with

1.1.1. Chvátal's geometric view

By definition a polyhedron P is integer if every face contains an integer point. By the integer Farkas lemma (see, for instance, [95] Corollary 4.1a) this in turn is equivalent to the fact that every supporting hyperplane contains an integer vector. The idea is now to look at every supporting hyperplane of $P = \{x \in \mathbb{R}^n_+:\ Ax = b\}$ and shift it closer to $P_I = \mathrm{conv}(X)$ until it contains an integer point.

Let $\{x \in \mathbb{R}^n:\ h^{\mathrm{T}}x = \vartheta\}$ be a supporting hyperplane of P with $P \subseteq \{x \in \mathbb{R}^n:\ h^{\mathrm{T}}x \leqslant \vartheta\}$ and h integer. Let

$$Q^1 := \bigcap_{(h,\vartheta)\in\theta} \{x \in \mathbb{R}^n:\ h^{\mathrm{T}}x \leqslant \lfloor \vartheta \rfloor\}, \tag{1}$$

where θ denotes the set of all supporting hyperplanes of P with integer left-hand side. Obviously, $P_I \subseteq Q^1$. At first sight it is not obvious that Q^1 is again a polyhedron, because there are infinitely many supporting hyperplanes. However, it turns out that Q^1 is again a polyhedron. This allows us to continue the process and apply the same procedure to Q^1. With

$$Q^0 := P \quad \text{and} \quad Q^{t+1} := (Q^t)^1$$

we have

$$P = Q^0 \supseteq Q^1 \supseteq \cdots \supseteq P_I.$$

Chvátal shows that P_I is obtained this way after a finite number of iterations when P is a polytope, and Schrijver shows the result when P is an arbitrary polyhedron.

Theorem 1.1 (Chvátal [30], Schrijver [94]). *Let P be a rational polyhedron. Then*
(i) *Q^1 is a polyhedron.*
(ii) *$Q^t = P_I$ for some finite t.*

The question remains how to generate hyperplanes on demand, i.e., how to find $(h,\vartheta)\in\theta$ that cuts off the current (fractional) solution of the LP relaxation $\min\{c^{\mathrm{T}}x\colon x\in P\}$. Gomory [46,48] gives an answer to this question.

1.1.2. Gomory's algorithmic view

Let x^* be an optimal solution of the LP relaxation $\min\{c^{\mathrm{T}}x\colon x\in P\}$, $P\subseteq R^{n_+}$ and $B\subseteq\{1,\dots,n\}$ be a basis of A with $x^*_B=A_B^{-1}b-A_B^{-1}A_Nx_N$ and $x^*_N=0$, where $N=\{1,\dots,n\}\setminus B$.

If x^* is integral, we terminate with an optimal solution for $\min\{c^{\mathrm{T}}x\colon x\in X\}$. Otherwise, one of the values x^*_B must be fractional. Let $i\in B$ be some index with $x^*_i\notin\mathbb{Z}$. Since every feasible integral solution $x\in X$ satisfies $x_B=A_B^{-1}b-A_B^{-1}A_Nx_N$,

$$A_{i\cdot}^{-1}b-\sum_{j\in N}A_{i\cdot}^{-1}A_{\cdot j}x_j\in\mathbb{Z}, \tag{2}$$

where $D_{i\cdot}$ and $D_{\cdot j}$ denotes the ith row and jth column of some matrix D, respectively. The term on the left remains integral when adding integer multiples of x_j, $j\in N$, or an integer to $A_{i\cdot}^{-1}b$. We obtain

$$f(A_{i\cdot}^{-1}b)-\sum_{j\in N}f(A_{i\cdot}^{-1}A_{\cdot j})x_j\in\mathbb{Z}, \tag{3}$$

where $f(\alpha)=\alpha-\lfloor\alpha\rfloor$, for $\alpha\in\mathbb{R}$. Since $0\leqslant f(\cdot)<1$ and $x\geqslant 0$, we conclude that

$$f(A_{i\cdot}^{-1}b)-\sum_{j\in N}f(A_{i\cdot}^{-1}A_{\cdot j})x_j\leqslant 0,$$

or equivalently,

$$\sum_{j\in N}f(A_{i\cdot}^{-1}A_{\cdot j})x_j\geqslant f(A_{i\cdot}^{-1}b) \tag{4}$$

is valid for P_I. Moreover, it is violated by the current linear programming solution x^*, since $x^*_N=0$ and $f(A_{i\cdot}^{-1}b)=f(x^*_i)>0$. After subtracting $x_i+\sum_{j\in N}A_{i\cdot}^{-1}A_{\cdot j}x_j=A_{i\cdot}^{-1}b$ from (4) we obtain

$$x_i+\sum_{j\in N}\lfloor A_{i\cdot}^{-1}A_{\cdot j}\rfloor x_j\leqslant\lfloor A_{i\cdot}^{-1}b\rfloor, \tag{5}$$

which is, when the right-hand side is not rounded, a supporting hyperplane with integer left-hand side, and thus a member of θ. Moreover, adding this inequality to the system $Ax=b$ preserves the property that all data are integral. Thus, the slack variable that is to be introduced for the new inequality can be required to be integer as well and the whole procedure can be iterated. In fact, Gomory [49] proves that with a particular choice of the generating row such cuts lead to a finite algorithm, i.e., after adding a finite number of inequalities, an integer optimal solution is found. Thus, it provides an alternative proof for Theorem 1.1.

Given P and a general point $x^*\in P$, the *separation problem* for Chvátal–Gomory inequalities is to determine whether $x^*\in P^1$, and if not to find an inequality $h^{\mathrm{T}}x\leqslant\lfloor\vartheta\rfloor$ cutting off x^*. An efficient procedure has been proposed when $h=uA$ with u restricted to be a $\{0,\frac{1}{2}\}$ vector [28], but the general problem has been shown to be NP-hard [108].

1.2. Mixed integer programs

The two approaches discussed so far fail when both integer and continuous variables are present. Chvátal's approach fails because the right-hand side cannot be rounded down in (1). Gomory's approach fails since it is no longer possible to add integer multiples to continuous variables to derive (3) from (2). For instance, $\frac{1}{3}+\frac{1}{3}x_1-2x_2\in\mathbb{Z}$ with $x_1\in\mathbb{Z}_+$, $x_2\in\mathbb{R}_+$ has a larger solution set than $\frac{1}{3}+\frac{1}{3}x_1\in\mathbb{Z}$. As a consequence, we cannot guarantee that the coefficients of the continuous variables are non-negative and therefore show the validity of (4). Nevertheless, it is possible to derive valid inequalities using the following *disjunctive argument*.

Observation 1.2. *Let* $(a^k)^{\mathrm{T}}x\leqslant\alpha^k$ *be a valid inequality for a polyhedron* $P^k\subseteq R^{n_+}$ *for* $k=1,2$. *Then,*

$$\sum_{i=1}^{n}\min(a_i^1,a_i^2)x_i\leqslant\max(\alpha^1,\alpha^2)$$

is valid for both $P^1\cup P^2$ *and* $\mathrm{conv}(P^1\cup P^2)$.

This observation applied in different ways yields valid inequalities for the mixed integer case. We present three methods that are all more or less based on Observation 1.2.

1.2.1. Gomory's mixed integer cuts

Consider again the situation in (2), where x_i, $i\in B$, is required to be integer. We use the following abbreviations $\bar{a}_j=A_{i\cdot}^{-1}A_{\cdot j}$, $\bar{b}=A_{i\cdot}^{-1}b$, $f_j=f(\bar{a}_j)$, $f_0=f(\bar{b})$, and $N^+=\{j\in N\colon \bar{a}_j\geqslant 0\}$ and $N^-=N\setminus N^+$. Expression (2) is equivalent to $\sum_{j\in N}\bar{a}_jx_j=f_0+k$ for some $k\in\mathbb{Z}$. We distinguish two cases, $\sum_{j\in N}\bar{a}_jx_j\geqslant 0$ and $\sum_{j\in N}\bar{a}_jx_j\leqslant 0$. In the first case,

$$\sum_{j\in N^+}\bar{a}_jx_j\geqslant f_0$$

must hold. In the second case, we have $\sum_{j\in N^-}\bar{a}_jx_j\leqslant f_0-1$, which is equivalent to

$$-\frac{f_0}{1-f_0}\sum_{j\in N^-}\bar{a}_jx_j\geqslant f_0.$$

Now we apply Observation 1.2 to the disjunction $P^1=P\cap\{x\colon \sum_{j\in N}\bar{a}_jx_j\geqslant 0\}$ and $P^2=P\cap\{x\colon \sum_{j\in N}\bar{a}_jx_j\leqslant 0\}$ and obtain the valid inequality

$$\sum_{j\in N^+}\bar{a}_jx_j-\frac{f_0}{1-f_0}\sum_{j\in N^-}\bar{a}_jx_j\geqslant f_0. \tag{6}$$

This inequality may be strengthened in the following way. Observe that the derivation of (6) remains unaffected when adding integer multiples to integer variables. By doing this we may put each integer variable either in the set N^+ or N^-. If a variable is in N^+, the final coefficient in (6) is $\bar{a}_j$ and thus the best possible coefficient after adding integer multiples is $f_j = f(\bar{a}_j)$. In N^- the final coefficient in (6) is $(f_0/(1-f_0))\bar{a}_j$ and thus $f_0(1-f_j)/(1-f_0)$ is the best choice. Overall, we obtain the best possible coefficient by using $\min(f_j, f_0(1-f_j)/(1-f_0))$. This yields Gomory's mixed integer cut [47]

$$\sum_{\substack{j:\ f_j \leqslant f_0 \\ j \text{ integer}}} f_j x_j + \sum_{\substack{j:\ f_j > f_0 \\ j \text{ integer}}} \frac{f_0(1-f_j)}{1-f_0} x_j$$

$$+ \sum_{\substack{j \in N^+ \\ j \text{ non-integer}}} \bar{a}_j x_j - \sum_{\substack{j \in N^- \\ j \text{ non-integer}}} \frac{f_0}{1-f_0} \bar{a}_j x_j \geqslant f_0. \tag{7}$$

Gomory [47] shows that an algorithm based on iteratively adding these inequalities solves $\min\{c^{\mathrm{T}}x\colon x \in X\}$ with $X = \{x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}\colon Ax = b\}$ in a finite number of steps provided $c^{\mathrm{T}}x \in \mathbb{Z}$ for all $x \in X$.

1.2.2. Mixed-integer-rounding cuts

Consider the following elementary mixed integer set $X = \{(x, y) \in \mathbb{Z} \times \mathbb{R}_+\colon x - y \leqslant b\}$ with $b \in \mathbb{R}$ and the inequality

$$x - \frac{1}{1-f(b)} y \leqslant \lfloor b \rfloor. \tag{8}$$

Proposition 1.3 (Nemhauser and Wolsey [78,79]). *Inequality* (8) *is valid for* $\operatorname{conv}(X)$.

Proof. Consider the disjunction $P^1 = X \cap \{(x, y)\colon x \leqslant \lfloor b \rfloor\}$ and $P^2 = X \cap \{(x, y)\colon x \geqslant \lfloor b \rfloor + 1\}$. For P^1 we immediately see that

$$(x - \lfloor b \rfloor)(1 - f(b)) \leqslant y$$

is valid by adding the inequalities $x - \lfloor b \rfloor \leqslant 0$ and $0 \leqslant y$ scaled with weights $1 - f(b)$ and 1. For P^2 we combine $-(x - \lfloor b \rfloor) \leqslant -1$ and $x - y \leqslant b$ with weights $f(b)$ and 1 to obtain

$$(x - \lfloor b \rfloor)(1 - f(b)) \leqslant y.$$

Thus, Observation 1.2 implies that $(x - \lfloor b \rfloor)(1 - f(b)) \leqslant y$ is valid for $\operatorname{conv}(P^1 \cup P^2) = \operatorname{conv}(X)$. $\square$

The basic observation expressed in Proposition 1.3 can now be extended to more general situations. Consider the following mixed integer set:

$$X = \{(x, y) \in \mathbb{Z}_+^n \times \mathbb{R}_+\colon a^{\mathrm{T}}x - y \leqslant b\},$$

with $a \in \mathbb{R}^n, b \in \mathbb{R}$. We take $f_i = f(a_i)$ and $f_0 = f(b)$ in the sequel.

Proposition 1.4 (Nemhauser and Wolsey [78,79]). *The inequality*

$$\sum_{i=1}^{n}\left(\lfloor a_i\rfloor + \frac{(f_i - f_0)^+}{1-f_0}\right)x_i - \frac{1}{1-f_0}y \leqslant \lfloor b\rfloor \tag{9}$$

is valid for $\operatorname{conv}(X)$, *where* $v^+ = \max(0,v)$ *for* $v\in\mathbb{R}$. *Inequality* (9) *is called a* mixed integer rounding (MIR) inequality.

Proof. Relax $a^{\mathrm{T}}x - y \leqslant b$ to $\sum_{i\in N^1}\lfloor a_i\rfloor x_i + \sum_{i\in N^2} a_i x_i - y \leqslant b$, where $N^1 = \{i\in\{1,\dots,n\}: f_i \leqslant f_0\}$ and $N^2 = \{1,\dots,n\}\setminus N^1$. Applying Proposition 1.3 to $w - z \leqslant b$ with $w = \sum_{i\in N^1}\lfloor a_i\rfloor x_i + \sum_{i\in N^2}\lceil a_i\rceil x_i \in\mathbb{Z}$ and $z = y + \sum_{i\in N^2}(1-f_i)x_i \geqslant 0$ yields

$$w - \frac{z}{1-f_0} \leqslant \lfloor b\rfloor. \tag{10}$$

Substituting w and z in (10) gives (9). □

MIR inequalities imply Gomory's mixed integer cuts (7) when applied to the mixed integer set $X = \{(x,y^-,y^+)\in\mathbb{Z}^n_+\times\mathbb{R}^2_+: a^{\mathrm{T}}x + y^+ - y^- = b\}$. To see this consider the relaxation $a^{\mathrm{T}}x - y^- \leqslant b$ of X. Proposition 1.4 gives

$$\sum_{i=1}^{n}\left(\lfloor a_i\rfloor + \frac{(f_i - f_0)^+}{1-f_0}\right)x_i - \frac{1}{1-f_0}y^- \leqslant \lfloor b\rfloor.$$

Subtracting the original inequality $a^{\mathrm{T}}x + y^+ - y^- = b$ gives Gomory's mixed integer cut (7).

Nemhauser and Wolsey [79] discuss MIR inequalities in a more general setting. They prove that MIR inequalities provide a complete description for any mixed 0–1 polyhedron. Marchand and Wolsey [69,70] show that certain strong cutting planes for structured mixed integer programs can be derived as MIR inequalities. They also show their computational effectiveness in solving general mixed integer programs.

1.2.3. Lift-and-project cuts

The idea of "lift and project" is to consider the integer programming problem, not in the original space, but in some space of higher dimension (lifting). Then inequalities found in this higher dimensional space are projected back to the original space resulting in tighter integer programming formulations. Versions of this approach differ in how the lifting and the projection are performed, see [10,66,96]. All approaches only apply to 0–1 mixed integer programming problems. We explain the ideas in [10] in more detail and show the connections and differences to [66,96].

The validity of the procedure is based on a trivial observation.

Observation 1.5. *If* $c_0 + c^{\mathrm{T}}x \geqslant 0$ *and* $d_0 + d^{\mathrm{T}}x \geqslant 0$ *are valid inequalities for* X, *then* $(c_0 + c^{\mathrm{T}}x)^{\mathrm{T}}(d_0 + d^{\mathrm{T}}x) \geqslant 0$ *is valid for* X.

Consider a 0–1 program $\min\{c^{\mathrm{T}}x: x\in X\}$ with $X = \{x\in\{0,1\}^p\times\mathbb{R}^{n-p}: Ax\leqslant b\}$, in which the system $Ax\leqslant b$ already contains the trivial inequalities $0\leqslant x_i\leqslant 1$ for $i = 1,\dots,p$. Let $P = \{x\in\mathbb{R}^n: Ax\leqslant b\}$ and $P_I = \operatorname{conv}(X)$. Consider the following procedure.

Algorithm 1.6 (Lift-and-Project).

1. *Select an index* $j \in \{1, \ldots, p\}$.
2. *Multiply* $Ax \leqslant b$ *by* x_j *and* $1 - x_j$ *giving*

$$(Ax)x_j \leqslant bx_j,$$

$$(Ax)(1 - x_j) \leqslant b(1 - x_j) \tag{11}$$

and substitute $y_i := x_i x_j$ *for* $i = 1, \ldots, n,\ i \neq j$ *and* $x_j := x_j^2$ (lifting). *Call the resulting polyhedron* $L_j(P)$.
3. *Project* $L_j(P)$ *back to the original space by eliminating variables* y_i. *Call the resulting polyhedron* P_j.

The following theorem shows that the jth component of each vertex of P_j is either zero or one.

Theorem 1.7 (Balas et al. [11]). $P_j = \operatorname{conv}(P \cap \{x \in \mathbb{R}^n\colon x_j \in \{0, 1\}\})$.

For any sequence of indices $i_1, \ldots, i_t \in \{1, \ldots, p\}, t \geqslant 1$ let

$$P_{i_1, i_2, \ldots, i_t} := (\ldots(P_{i_1})_{i_2} \ldots)_{i_t}.$$

A repeated application of Algorithm 1.6 yields P_I.

Theorem 1.8 (Balas et al. [10]). $P_{i_1, \ldots, i_t} = \operatorname{conv}(P \cap \{x \in \mathbb{R}^n\colon x_{i_h} \in \{0, 1\},\ h = 1, \ldots, t\})$.

Theorem 1.8 shows that the result does not depend on the order in which one applies Algorithm 1.6 to the selected variable. Thus, we may write $P_{\{i_1, \ldots, i_t\}}$ instead of $P_{i_1, \ldots, i_t}$ and $P_{\{1, \ldots, p\}} = P_I$.

The problem that remains in order to implement Algorithm 1.6 is to carry out Step 3. Let $L_j(P) = \{(x, y)\colon Dx + By \leqslant d\}$. Then the projection of $L_j(P)$ onto the x-space can be described by

$$P_j = \{x\colon (u^{\mathrm{T}} D)x \leqslant u^{\mathrm{T}} d \text{ for all } u \in C\},$$

where $C = \{u\colon u^{\mathrm{T}} B = 0,\ u \geqslant 0\}$. Thus, the problem of finding a valid inequality in Step 3 of Algorithm 1.6 that cuts off a current (fractional) solution x^* can be solved by the linear program

$$\begin{aligned} &\max\ u^{\mathrm{T}}(Dx^* - d), \\ &\quad u \in C. \end{aligned} \tag{12}$$

This linear program is unbounded, if there is a violated inequality, since C is a polyhedral cone. For algorithmic convenience C is often truncated by some “normalizing set”, see [10]. If an integer variable x_j that attains a fractional value in a basic feasible solution is used to determined the index j in Algorithm 1.6, then an optimal solution to (11) indeed cuts off x^*.

The computational merits of lift-and-project cuts to solve real-world problems are discussed in [10,11].

There is a close connection between the lift-and-project method and disjunctive programming. In fact, Theorem 1.7 states that $P_j = \text{conv}(P^0 \cup P^1)$ where $P^0{:=}P \cap \{x \in \mathbb{R}^n\colon x_j = 0\}$ and $P^1{:=}P \cap \{x \in \mathbb{R}^n\colon x_j = 1\}$. The inequalities obtained by projecting $L_j(P)$ onto the x-space may be viewed as inequalities obtained from the disjunction of P into P^0 and P^1. Thus, lift-and-project is a specialization of disjunctive programming, see, for instance, [8,60] for further details on this issue.

Observation 1.5 can be applied to a more general setting. For the ease of exposition we assume that our mixed integer program is indeed a pure integer program, i.e., $p = n$. Sherali and Adams [96] suggest lifting the problem to a higher dimensional space by multiplying $Ax \leqslant b$ by every product $(\prod_{j\in J_1} x_j)(\prod_{j\in J_2}(1 - x_j))$ such that $J_1, J_2 \subseteq \{1,\dots,n\}$ are disjoint and $|J_1 \cup J_2| = d$ for some fixed value $d \in \{1,\dots,n\}$. They linearize the problem by setting $x_i = x_i^k$, $2 \leqslant k \leqslant d+1$, and by replacing every product $\prod_{j\in J} x_j$ by a single variable y_J for $J \subseteq \{1,\dots,n\}$. Thereafter, the high-dimensional problem is projected to the space of x-variables. If $d = n$ is chosen, then this procedure directly yields a linear description of P_I.

Setting $d = 1$, the first step of the above procedure leads to the system

$$(Ax)x_j \leqslant bx_j \quad \text{for } j = 1,\dots,n,$$

$$(Ax)(1 - x_j) \leqslant b(1 - x_j) \quad \text{for } j = 1,\dots,n.$$

Setting $y_{ij} = x_i x_j$ for $1 \leqslant i < j \leqslant n$, and then projecting back to the original space leads to a polyhedron $N(P) \subseteq \bigcap_{j=1}^n P_j$. It is clear from Theorem 1.7 that this tighter procedure must be repeated at most n times to terminate with P_I. Lovász and Schrijver [66] studied this projection in more detail. They note that if $x_0 = 1$, then the product of two valid inequalities

$$(c_0 + c^{\mathrm{T}}x)^{\mathrm{T}}(d_0 + d^{\mathrm{T}}x) = c^{\mathrm{T}} \begin{pmatrix} x_0 \\ x \end{pmatrix} (x_0, x^{\mathrm{T}})d = c^{\mathrm{T}}Xd \geqslant 0,$$

where $X = \binom{x_0}{x}(x_0, x^{\mathrm{T}})$ is a symmetric and positive semidefinite matrix. This is pursued in Section 2.3.

We want to emphasize here that in contrast to the pure integer case none of the cutting plane procedures presented yields a finite algorithm for general mixed integer programs. Gomory needs an integer restricted objective function, and the other two provide finiteness only for 0–1 mixed integer programs. Cook, Kannan, and Schrijver [33] present the so-called split cuts. These cuts are again based on Observation 1.2 and may be viewed as special disjunctive cuts. They turn out to be equivalent to MIR inequalities [79]. However, Cook et al., show that the split cuts in combination with a certain rounding technique, which is based on the idea of discretizing the continuous variables, suffice to generate the mixed integer hull of a polyhedron. See also [83].

2. Simple structures

Above we have looked at valid inequalities for IPs and MIPs. If we restrict our attention to a single constraint, or a small subset of constraints, even a general problem may exhibit some "local" structure. For example, all variables appearing in a constraint may be 0–1 variables, or a small part of the MIP may be a network flow problem. Here, we look at ways to obtain stronger inequalities by using such local structure.

2.1. Knapsacks and cover inequalities

The concept of a cover has been used extensively in the literature to derive valid inequalities for (mixed) integer sets. In this section, we first show how to use this concept to derive cover inequalities for the 0–1 knapsack set. We then discuss how to extend these inequalities to more complex mixed integer sets.

Consider the 0–1 knapsack set

$$K = \left\{ x \in \{0,1\}^N : \sum_{j \in N} a_j x_j \leqslant b \right\}$$

with non-negative coefficients, i.e., $a_j \geqslant 0$ for $j \in N$ and $b \geqslant 0$. The set $C \subseteq N$ is a *cover* if

$$\lambda = \sum_{j \in C} a_j - b > 0. \tag{13}$$

In addition, the cover C is said to be *minimal* if $a_j \geqslant \lambda$ for all $j \in C$. To each cover C, we can associate a simple valid inequality which states that "not all variables x_j for $j \in C$ can be set to one simultaneously".

Proposition 2.1 (Balas [7], Hammer et al. [57], Padberg [85], Wolsey [103]). *Let $C \subseteq N$ be a cover. The cover inequality*

$$\sum_{j \in C} x_j \leqslant |C| - 1 \tag{14}$$

is valid for K. Moreover, if C is minimal, then the inequality (14) *defines a facet of* $\mathrm{conv}(K_C)$ *where* $K_C = K \cap \{x \colon x_j = 0,\ j \in N \setminus C\}$.

Example 2.2. Consider the 0–1 knapsack set

$$K = \{x \in \{0,1\}^6 \colon 5x_1 + 5x_2 + 5x_3 + 5x_4 + 3x_5 + 8x_6 \leqslant 17\}.$$

$C = \{1,2,3,4\}$ is a minimal cover for K and the corresponding cover inequality

$$x_1 + x_2 + x_3 + x_4 \leqslant 3$$

defines a facet of $\mathrm{conv}(\{x \in \{0,1\}^4 \colon 5x_1 + 5x_2 + 5x_3 + 5x_4 \leqslant 17\})$.

If a cover C is not minimal, then it is easily seen that the corresponding cover inequality is redundant, i.e., it is the sum of a minimal cover inequality and some upper bound constraints.

As described in the next subsection, lifting can be used to strengthen cover inequalities and to obtain a large class of facet-defining inequalities for $\mathrm{conv}(K)$ called *lifted cover inequalities*. Generalizations of cover inequalities can be found in [43,99,106] where the polyhedral structures of, respectively, the 0–1 knapsack set with generalized upper bounds constraints, the 0–1 knapsack with precedence constraints and the multiple 0–1 knapsack set are studied. Lifted cover inequalities have been used successfully in general purpose branch-and-cut algorithms to tighten the formulation of 0–1 integer programs [36,54]. In [13], it is shown how minimal covers, lifting and complementation (replacing the binary variable x_j by its complement $1-\bar{x}_j$) can be used to obtain all the non-trivial facets of the 0–1 integer programming polytope with positive coefficients.

The concept of cover is also useful in the study of the polyhedral structure of problems containing both 0–1, integer and continuous variables. Consider the mixed 0–1 knapsack set

$$S=\left\{(x,s)\in\{0,1\}^N\times\mathbb{R}_+:\ \sum_{j\in N}a_jx_j\leqslant b+s\right\}$$

with non-negative coefficients, i.e., $a_j\geqslant 0$ for $j\in N$ and $b\geqslant 0$.

Proposition 2.3 (Marchand and Wolsey [71]). *Let $C\subseteq N$ be a cover, i.e., C is a subset of N satisfying* (13). *The inequality*

$$\sum_{j\in C}\min(a_j,\lambda)x_j\leqslant\sum_{j\in C}\min(a_j,\lambda)-\lambda+s \tag{15}$$

is valid for S. Moreover, the inequality (15) *defines a facet of* $\operatorname{conv}(S_C)$ *where* $S_C=S\cap\{x:\ x_j=0, j\in N\setminus C\}$.

Note here that each cover C gives rise to a cover inequality that defines a facet of $\operatorname{conv}(S_C)$. This is in contrast to the pure integer case where only minimal covers induce facets.

Example 2.4. Consider the mixed 0–1 knapsack set

$$S=\{(x,s)\in\{0,1\}^6\times\mathbb{R}_+:\ 5x_1+5x_2+5x_3+5x_4+3x_5+8x_6\leqslant 17+s\}.$$

Taking $C'=\{1,2,3,6\}$ a (non-minimal) cover for S, the associated cover inequality

$$5x_1+5x_2+5x_3+6x_6\leqslant 15+s$$

defines a facet of $\operatorname{conv}(\{(x,s)\in\{0,1\}^4\times\mathbb{R}_+:\ 5x_1+5x_2+5x_3+8x_6\leqslant 17+s\})$.

Cover inequalities of the form (15) can be used to derive valid inequalities for more complex mixed integer sets. We illustrate this observation by showing how to derive valid inequalities for an elementary flow model consisting of inflow arcs with capacities and fixed costs, and a constraint on the total inflow.

Consider the (flow) set

$$X = \left\{ (x,y) \in \{0,1\}^N \times \mathbb{R}_+^N : \sum_{j \in N} y_j \leqslant b, y_j \leqslant a_j x_j, j \in N \right\}$$

and let $C \subseteq N$ be a (flow) cover, i.e., C is a subset of N satisfying (13). In $\sum_{j \in N} y_j \leqslant b$, ignore y_j for $j \in N \setminus C$ and replace y_j by $a_j x_j - s_j$ for $j \in C$ where $s_j \geqslant 0$ is a slack variable. We obtain

$$\sum_{j \in C} a_j x_j \leqslant b + \sum_{j \in C} s_j.$$

Using Proposition 2.3, we have that the following inequality is valid for X

$$\sum_{j \in C} \min(a_j, \lambda) x_j \leqslant \sum_{j \in C} \min(a_j, \lambda) - \lambda + \sum_{j \in C} s_j,$$

or equivalently, substituting $a_j x_j - y_j$ for s_j,

$$\sum_{j \in C} [y_j + (a_j - \lambda)^+ (1 - x_j)] \leqslant b.$$

Proposition 2.5 (Padberg et al. [86]). *Let $C \subseteq N$ be a flow cover (C is a subset of N satisfying (13)) with $\max_{j \in C} a_j > \lambda$. The flow cover inequality*

$$\sum_{j \in C} [y_j + (a_j - \lambda)^+ (1 - x_j)] \leqslant b \tag{16}$$

is a facet-defining inequality for $\mathrm{conv}(X)$.

Flow models have been extensively studied in the literature. Various generalizations of the flow cover inequality (16) have been derived for more complex flow models. In [100], a family of flow cover inequalities is described for a general single node flow model containing variable lower and upper bounds. Generalizations of flow cover inequalities to lot-sizing and capacitated facility location problems can also be found, respectively, in [2,87]. Flow cover inequalities have been used successfully in general purpose branch-and-cut algorithms to tighten formulations of mixed integer sets [52,53,101]. See Example 2.8 and Section 3.

Cover inequalities appear also in other contexts. In [29] cover inequalities are derived for the knapsack set with general integer variables. Unfortunately, in this case, the resulting inequalities do not define facets of the convex hull of the knapsack set restricted to the variables defining the cover. More recently, the notion of cover has been used to define families of valid inequalities for the complementarity knapsack set [39].

2.2. Lifting

The lifting technique is a general approach that has been used in a wide variety of contexts to strengthen valid inequalities. For simplicity of exposition, we first illustrate the main concepts related to this technique by lifting binary variables in a 0–1 knapsack set.

Consider the 0–1 knapsack set

$$K = \left\{ x \in \{0,1\}^N \colon \sum_{j \in N} a_j x_j \leqslant b \right\}$$

and let M be a subset of N. Suppose that we have an inequality,

$$\sum_{j \in M} \pi_j x_j \leqslant \pi_0, \tag{17}$$

which is valid for $K_M = K \cap \{x \colon x_j = 0, j \in N \setminus M\}$. The *lifting problem* is to find the lifting coefficients $\{\pi_j\}_{j \in N \setminus M}$ so that

$$\sum_{j \in N} \pi_j x_j \leqslant \pi_0 \tag{18}$$

is valid for K. Ideally, we would like inequality (18) to be "strong" (i.e., if inequality (17) defines a face of high dimension of $\operatorname{conv}(K_M)$, we would like the inequality (18) to define a face of high dimension of $\operatorname{conv}(K)$).

2.2.1. Sequential lifting

One way of obtaining coefficients $\{\pi_j\}_{j \in N \setminus M}$ is to apply *sequential lifting*: lifting coefficients π_j are evaluated one after another. More specifically, the coefficient π_k is computed for a given $k \in N \setminus M$ so that

$$\pi_k x_k + \sum_{j \in M} \pi_j x_j \leqslant \pi_0 \tag{19}$$

is valid for $K_{M \cup \{k\}}$. This can be done by considering the *lifting function*

$$\Phi_M(u) = \min \left\{ \pi_0 - \sum_{j \in M} \pi_j x_j \colon \sum_{j \in M} a_j x_j \leqslant b - u, x \in \{0,1\}^{|M|} \right\}. \tag{20}$$

Proposition 2.6 (Sequential lifting [85]). *Suppose $K_{M \cup \{k\}} \cap \{x \colon x_k = 1\} \neq \emptyset$. Inequality* (19) *is valid for $K_{M \cup \{k\}}$ if $\pi_k \leqslant \Phi_M(a_k)$. Moreover, if $\pi_k = \Phi_M(a_k)$ and* (17) *defines a face of dimension t of* $\operatorname{conv}(K_M)$, *then* (19) *defines a face of* $\operatorname{conv}(K_{M \cup \{k\}})$ *of at least dimension $t+1$.*

If one now intends to lift a second variable, then it becomes necessary to update the function Φ_M. Specifically, if $k \in N \setminus M$ was introduced first with a lifting coefficient π_k, then the lifting function becomes

$$\Phi_{M \cup \{k\}}(u) = \min \left\{ \pi_0 - \sum_{j \in M \cup \{k\}} \pi_j x_j \colon \sum_{j \in M \cup \{k\}} a_j x_j \leqslant b - u, x \in \{0,1\}^{|M|+1} \right\},$$

so in general, function Φ_M can decrease as more variables are lifted in. As a consequence, lifting coefficients depend on the order in which variables are lifted and therefore different lifting sequences often lead to different valid inequalities.

Example 2.7. Consider the 0–1 knapsack set

$$K = \{x \in \{0,1\}^6\colon 5x_1 + 5x_2 + 5x_3 + 5x_4 + 3x_5 + 8x_6 \leqslant 17\}$$

and let $M = \{1,2,3,4\}$. The inequality

$$x_1 + x_2 + x_3 + x_4 \leqslant 3$$

is valid for $K_{\{1,2,3,4\}}$. Lifting variable x_5 and then variable x_6 leads to

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leqslant 3.$$

However, lifting variable x_6 and then variable x_5 leads to

$$x_1 + x_2 + x_3 + x_4 + 2x_6 \leqslant 3.$$

It can be checked that both inequalities define facets of $\mathrm{conv}(K)$.

One of the key questions to be dealt with when implementing such a lifting approach is how to compute lifting coefficients π_j. To perform "exact" sequential lifting (i.e., to compute at each step the lifting coefficient given by the lifting function), we have to solve a sequence of integer programs. In the case of the lifting of variables for the 0–1 knapsack set this can be done efficiently using a dynamic programming approach related to the following recursion formula:

$$\Phi_{M\cup\{k\}}(u) = \min[\Phi_M(u), \Phi_M(u + a_k) - \Phi_M(a_k)].$$

Using such a lifting approach, facet-defining inequalities for the 0–1 knapsack set have been derived [14,7,57,103,85] and embedded in a branch-and-bound framework to solve to optimality particular types of 0–1 integer programs [36].

We now indicate how the lifting ideas can be extended to treat variables fixed to values other than zero, and to handle more than one variable at a time.

Lifting a binary variable fixed to one

Consider the binary knapsack set

$$K_{M\cup\{k\}} = \left\{ x \in \{0,1\}^{|M|+1}\colon \sum_{M\cup\{k\}} a_j x_j \leqslant b + a_k \right\}.$$

Note that with $x_k = 1$, this reduces to the set K_M for which a (facet-defining) inequality $\sum_{j\in M} \pi_j x_j \leqslant \pi_0$ of $\mathrm{conv}(K_M)$ is given. So here we ask for what values of π_k, the inequality

$$\sum_{j\in M} \pi_j x_j + \pi_k(1 - x_k) \leqslant \pi_0$$

is valid for $K_{M\cup\{k\}}$.

The inequality is valid by construction when $x_k = 1$, and when $x_k = 0$, it is valid if and only if $\pi_k \leqslant \Phi_M(-a_k)$. It follows that

$$\sum_{j\in M} \pi_j x_j - \Phi_M(-a_k)x_k \leqslant \pi_0 - \Phi_M(-a_k)$$

is facet-defining for $\mathrm{conv}(K_{M\cup\{k\}})$. Note that an alternative way to derive this inequality is to work with the complemented variable $\bar{x}_k = 1 - x_k$, which is fixed to zero and then lifted.

Lifting a variable upper bound pair fixed to zero
Consider the set

$$X_{M\cup\{k\}} = \left\{(x,y)\in\{0,1\}^{|M|+1}\times\mathbb{R}_+^{|M|+1}: \sum_{M\cup\{k\}} y_j \leqslant b, y_j \leqslant a_j x_j, j\in M\cup\{k\}\right\}.$$

Note that with $(x_k, y_k) = (0,0)$, this reduces to the flow set over

$$X_M = \left\{(x,y)\in\{0,1\}^M\times\mathbb{R}_+^M: \sum_{j\in M} y_j \leqslant b, y_j \leqslant a_j x_j, j\in M\right\}.$$

Now suppose that the inequality

$$\sum_{j\in M}\pi_j x_j + \sum_{j\in M}\mu_j y_j \leqslant \pi_0$$

is valid and facet-defining for $\mathrm{conv}(X_M)$.

As before, let

$$\Psi_M(u) = \min\left\{\pi_0 - \sum_{j\in M}\pi_j x_j - \sum_{j\in M}\mu_j y_j: \sum_{j\in M} y_j \leqslant b-u,\right.$$
$$\left. y_j \leqslant a_j x_j, j\in M, (x,y)\in\{0,1\}^{|M|}\times\mathbb{R}_+^{|M|}\right\}.$$

Now the inequality

$$\sum_{j\in M}\pi_j x_j - \sum_{j\in M}\mu_j y_j + \pi_k x_k + \mu_k y_k \leqslant \pi_0$$

is valid if and only if $\pi_k + \mu_k u \leqslant \Psi_M(u)$ for all $0 \leqslant u \leqslant a_k$, ensuring that all the feasible points with $(x_k, y_k) = (1, u)$ satisfy the inequality.

So the inequality defines a facet if the affine function $\pi_k + \mu_k u$ lies below the function $\Psi_M(u)$ in the interval $[0, a_k]$ and touches it in two points different from $(0,0)$, thereby increasing the number of affinely independent tight points by the number of new variables. In [53] it is also shown how to lift the pair (x_k, y_k) when y_k has been fixed to a_k and x_k to 1.

Example 2.8. (i) Consider the 0–1 knapsack set

$$K = \{x\in\{0,1\}^5: 5x_1 + 5x_2 + 5x_3 + 5x_4 + 7x_5 \leqslant 22\}.$$

Fixing x_5=1, we obtain as before that $\sum_{j=1}^4 x_j \leqslant 3$ is facet-defining for $\mathrm{conv}(K_{\{1,2,3,4\}})$. As $\Phi(-7) = 3 - 4 = -1$,

$$x_1 + x_2 + x_3 + x_4 + x_5 \leqslant 4$$

is facet-defining for $\mathrm{conv}(K)$.

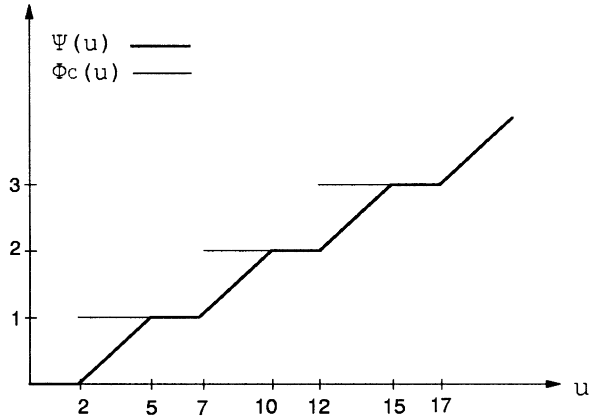

Fig. 1. Functions $\Phi_C(u)$ and $\Psi(u)$.

(ii) Consider the flow set

$$X' = \{(x,y) \in \{0,1\}^5 \times \mathbb{R}^5_+ : \sum_{j=1}^{5} y_j \leqslant 17, y_j \leqslant 5x_j, j = 1,\ldots,4, y_5 \leqslant 8x_5\}.$$

Fixing $(x_5, y_5) = (0,0)$, the flow cover inequality (16)

$$y_1 + y_2 + y_3 + y_4 - 2x_1 - 2x_2 - 2x_3 - 2x_4 \leqslant 9$$

is facet-defining for the resulting set $\operatorname{conv}(X_M)$.

The function $\Psi_M(u)$ is readily seen to satisfy

$$\Psi_M(u) = 0 \quad \text{for } 0 \leqslant u \leqslant 2,$$

$$\Psi_M(u) = u - 2 \quad \text{for } 2 \leqslant u \leqslant 5,$$

$$\Psi_M(u) = 3 \quad \text{for } 5 \leqslant u \leqslant 7,$$

$$\Psi_M(u) = 3 + (u - 7) \quad \text{for } 7 \leqslant u \leqslant 10, \text{ etc.}$$

Now for $(\alpha,\beta) = (0,0), (\alpha,\beta)(1,1)^{\mathrm{T}} = \Psi_M(1)$ and $(\alpha,\beta)(1,2)^{\mathrm{T}} = \Psi_M(2)$.

For $(\alpha,\beta) = (-\frac{6}{5}, \frac{3}{5}), (\alpha,\beta)(1,2)^{\mathrm{T}} = \Psi_M(2)$ and $(\alpha,\beta)(1,7)^{\mathrm{T}} = \Psi_M(7)$.

For $(\alpha,\beta) = (-4,1), (\alpha,\beta)(1,7)^{\mathrm{T}} = \Psi_M(7)$ and $(\alpha,\beta)(1,8)^{\mathrm{T}} = \Psi_M(8)$.

So three facet-defining inequalities

$$y_1 + y_2 + y_3 + y_4 - 2x_1 - 2x_2 - 2x_3 - 2x_4 \leqslant 9,$$

$$y_1 + y_2 + y_3 + y_4 - 2x_1 - 2x_2 - 2x_3 - 2x_4 + \tfrac{3}{5}y_5 - \tfrac{6}{5}x_5 \leqslant 9$$

and

$$y_1 + y_2 + y_3 + y_4 - 2x_1 - 2x_2 - 2x_3 - 2x_4 + y_5 - 4x_5 \leqslant 9$$

are obtained for $\operatorname{conv}(X')$. In Fig. 1 the function $\Psi = \lambda\Psi_M$ is shown.

In theory, "exact" sequential lifting can be applied to derive valid inequalities for any kind of mixed integer set. However, in practice, this approach is only useful to generate valid inequalities for sets for which one can associate a lifting function that can be evaluated efficiently.

Lifting is applied in the context of set packing problems to obtain facets from odd-hole inequalities [84], see Section 2.4. Other uses of sequential lifting can be found in [29] where the lifting of continuous and integer variables is used to extend the class of lifted cover inequalities to a mixed knapsack set with general integer variables. In [72,73] lifting is used to define (lifted) feasible set inequalities for an integer set defined by multiple integer knapsack constraints, see Section 2.3.

Sequential lifting is not the only way of computing lifting coefficients. We now discuss a general approach in which an "a priori" characterization is used to compute lifting coefficients.

2.2.2. Sequence independent lifting and superadditivity

Returning to the $0-1$ knapsack set K, we show how to evaluate lifting coefficients $\{\pi_j\}_{j\in N\setminus M}$ when we want to lift all variables in $N \setminus M$ *simultaneously.*

Because the function Φ_M may decrease as more variables are lifted in, taking $\{\Phi_M(a_j)\}_{j\in N\setminus M}$ as lifting coefficients does not in general lead to a valid inequality for K. Therefore to obtain a "sequence independent lifting", we have to find a function $\Psi:\mathbb{R}\to\mathbb{R}$ with $\Psi(u)\leqslant \Phi_M(u)$ so that

$$\sum_{j\in N\setminus M}\Psi(a_j)x_j+\sum_{j\in M}\pi_j x_j\leqslant \pi_0 \tag{21}$$

is valid for K. In the next proposition we characterize such a function Ψ. We first introduce a definition.

Definition 2.9. A function $F:\mathbb{R}\to\mathbb{R}$ is superadditive on $\mathbb{R}$ if $F(d_1)+F(d_2)\leqslant F(d_1+d_2)$ for all $d_1,d_2\in\mathbb{R}$.

Proposition 2.10. *Sequence independent lifting* [52,104]. *Let* $\Psi:\mathbb{R}\to\mathbb{R}$ *be a function. If* (i) $\Psi(u)\leqslant\Phi_M(u)$ *for all* $u\in\mathbb{R}$ *and* (ii) $\Psi(u)$ *is superadditive on* $\mathbb{R}$, *then inequality* (21) *is valid for* K.

Condition (ii) is quite restrictive. However, by considering the lifting of variables whose coefficients in the knapsack constraint take particular values, one can relax assumption (ii). In particular, if we suppose that all coefficients a_j are positive, condition (ii) becomes $\Psi(u)$ is superadditive on $\mathbb{R}_+$. We now illustrate this idea by deriving particular lifted cover inequalities using a superadditive function.

Consider a 0–1 knapsack set K in which $a_j>0$ for all $j\in N$. If $C\subseteq N$ is a minimal cover, the cover inequality

$$\sum_{j\in C}x_j\leqslant |C|-1$$

is valid for $K_C = K \cap \{x\colon x_j = 0, j \in N \setminus C\}$. The lifting function here is

$$\Phi_C(u) = \min\left\{ |C| - 1 - \sum_{j\in C} x_j \;\middle|\; \sum_{j\in C} a_j x_j \leqslant b - u, x \in \{0,1\}^{|C|} \right\}.$$

Suppose $C = \{1,\ldots,r\}$ and $a_j \geqslant a_{j+1}$ for all $j \in \{1,\ldots,r-1\}$. Let $A_j = \sum_{t=1}^{j} a_t$ and let $A_0 = 0$. The function

$$\Psi(u) = \begin{cases} j & \text{if } A_j \leqslant u \leqslant A_{j+1} - \lambda \text{ for } j = 0,\ldots,r-1, \\ j + [u - A_j]/\lambda & \text{if } A_j - \lambda \leqslant u \leqslant A_j \quad \text{for } j = 1,\ldots,r-1, \\ r + [u - A_r]/\lambda & \text{if } A_r - \lambda \leqslant u \end{cases}$$

is dominated by $\Phi_C(u)$ and is superadditive on $\mathbb{R}_+$. Therefore

$$\sum_{j\in N\setminus C} \Psi(a_j) x_j + \sum_{j\in C} x_j \leqslant |C| - 1 \tag{22}$$

is valid for K.

Example 2.7 (continued). The inequality (22) associated to $C = \{1,2,3,4\}$ is

$$x_1 + x_2 + x_3 + x_4 + \tfrac{1}{3}x_5 + \tfrac{4}{3}x_6 \leqslant 3.$$

The functions $\Phi_C(u)$ and $\Psi(u)$ are shown in Fig. 1.

Again sequence independent lifting can be extended to the lifting of valid inequalities for more general mixed integer sets [52]. In [53], simultaneous lifting of pairs of variables (included in the same variable upper bound constraint) is studied. Sequence independent lifted flow cover inequalities are obtained. In some of the cases studied there, the lifting function itself is shown to be superadditive. In [71], classes of facet-defining inequalities for the mixed knapsack set are obtained using the superadditivity of the lifting function first on $\mathbb{R}_+$ and then on $\mathbb{R}_-$, i.e., first lifting variables with positive coefficients, and then those with negative coefficients.

Other uses of lifting can be found in the literature. In [10,11], lift-and-project cuts are generated in the space of the fractional variables. The cutting planes are then lifted in the full space of variables. Lifting in this approach plays a central role because it reduces the computational effort required to generate lift-and-project cuts. A similar idea is used in [3] where cutting planes for the symmetric travelling salesman problem are generated from a polytope obtained by projection onto a small subset of the original variables.

2.3. Knapsacks and feasible set inequalities

Section 1.1 showed a way to derive an elementary inequality by forbidding an infeasible subset of items of a 0–1 knapsack set. We now investigate a way of defining valid inequalities for the 0–1 knapsack set starting with a feasible set and again using sequential lifting. This yields a generalization of the cover inequalities.

Consider again the 0–1 knapsack set

$$K = \left\{ x \in \{0,1\}^N : \sum_{j \in N} a_j x_j \leqslant b \right\}$$

with $a_j > 0$ for $j \in N$.

Let $T \subseteq N$ be a feasible set, i.e., $\sum_{j \in T} a_j \leqslant b$ and $w : T \to \mathbb{Z}_+ \setminus \{0\}$ a weighting of the items in T. We denote the slack by $r = b - \sum_{j \in T} a_j \geqslant 0$. Clearly, the inequality $\sum_{i \in T} w_i x_i \leqslant w(T)$ is valid for $K \cap \{x: x_i = 0 \text{ for } i \in N \setminus T\}$. Then we carry out sequential lifting as in the previous section.

Proposition 2.11 (Weismantel [102]). *If T is a feasible set and $w : T \to \mathbb{Z}_+ \setminus \{0\}$, the inequality*

$$\sum_{i \in T} w_i x_i + \sum_{j \in N \setminus T} \pi_j x_j \leqslant w(T)$$

is valid for K, where $(\mu_1, \ldots, \mu_{n-|T|})$ is a permutation of $N \setminus T$, Φ_T is the lifting function (20) *with $\pi_j = w_j$ for $j \in T$ and $\pi_0 = w(T)$, and $\pi_{\mu_i} = \Phi_{T \cup \{\mu_1, \ldots, \mu_{i-1}\}}(a_{\mu_i})$.*

We observe that if $w_i = 1$ for all $i \in T$, then $\pi_0 = |T|$ and

$$\Phi_T(u) = \min \left\{ |S| : S \subseteq T, \sum_{j \in S} a_j \geqslant u - r \right\}.$$

It follows immediately in this case that $\Phi_T(u) = 0$ for $0 \leqslant u \leqslant r$, and thus $\pi_j = 0$ whenever $j \in N \setminus T$ and $a_j \leqslant r$.

Example 2.12. Consider the knapsack polytope conv(K) defined as the convex hull of all 0–1 vectors that satisfy the constraint

$$3x_1 + 4x_2 + 6x_3 + 7x_4 + 9x_5 + 18x_6 \leqslant 21.$$

Taking the feasible set $T = \{1,2,3,4\}$, we obtain a slack $r = 1$. Choosing the permutation (5,6), we obtain coefficients $\pi_5 = 2$ and $\pi_6 = 3$. The resulting feasible set inequality

$$x_1 + x_2 + x_3 + x_4 + 2x_5 + 3x_6 \leqslant 4$$

defines a facet of conv(K).

Feasible set inequalities associated with a set T and weights $w_i = 1$ for all $i \in T$ subsume the family of lifted cover and $(1,k)$-configuration inequalities. Specifically, a set $T \cup \{z\} \subseteq N$ with $\sum_{i \in T} a_i \leqslant b$ is called a $(1,k)$-configuration, if every k-element subset of T together with the element z forms a minimal cover. This configuration gives rise to a valid inequality for K,

$$\sum_{i \in T} x_i + (|T| - k + 1)x_z \leqslant |T|.$$

It is a characteristic of feasible set inequalities that lifting coefficients can be computed in polynomial time under modest assumptions on the weights w_i of the items $i \in T$, see [102]. Indeed, the exact lifting coefficient of an item either equals a certain lower bound or equals this lower bound plus one. This generalizes an earlier result where this property was shown to hold for the lifting of minimal cover inequalities [14].

Theorem 2.13. *For $i \in N \setminus T$ with $a_i > r$, the coefficient π_i in any feasible set inequality associated with T and the weights $w_i = 1$ for all $i \in T$ satisfies*

$$\Phi_T(a_i) - 1 \leqslant \pi_i \leqslant \Phi_T(a_i).$$

In fact, Theorem 2.13 extends to more general families of feasible set inequalities where the coefficients of the items in the feasible set are not restricted to the value one, see [102]. Another extension of feasible set inequalities in [72,73] applies to general integer programs.

2.4. 0–1 matrices and valid inequalities

Integer and mixed integer programs often contain some constraints with only 0–1 coefficients. In addition, many preprocessors for integer programs automatically generate logical inequalities of the form $x_i + x_j \leqslant 1, x_i \leqslant x_j$, cover inequalities, etc. This naturally leads to the study of integer programs with 0–1 matrices.

The study of such problems, and in particular the set packing and covering problems, plays a prominent role in combinatorial optimization. These problems are among the most studied with a beautiful theory involving topics such as perfect, ideal, or balanced matrices, perfect graphs, the theory of blocking and anti-blocking polyhedra, independence systems and semidefinite programming.

The focus of this section is on a (partial) description of the associated polyhedra by means of inequalities. Assuming that relaxations of various integer programs yield set packing/covering problems, knowledge about these polyhedra can be used to strengthen the formulation of the original problem.

Definition 2.14. Let $A \in \{0,1\}^{m \times n}$ be a 0–1 matrix and $c \in \mathbb{R}^n$. The 0–1 integer programs

$$\max \{c^{\mathrm{T}}x\colon Ax \leqslant \mathbf{1}, x \in \{0,1\}^n\}, \tag{23}$$

$$\min \{c^{\mathrm{T}}x\colon Ax \geqslant \mathbf{1}, x \in \{0,1\}^n\} \tag{24}$$

are called the *set packing* and *set covering problems*, respectively.

Each column j of A can be viewed as the incidence vector of a subset F_j of the ground set $\{1,\dots,m\}$, i.e., $F_j := \{i \in \{1,\dots,m\}\colon A_{ij} = 1\}$. With this interpretation, the set packing problem consists of finding a collection of sets from $F_1,\dots,F_n$ that are mutually disjoint and maximal with respect to the objective function c. Analogously, the covering problem aims at finding a collection of subsets whose union yields the ground set and is minimal with respect to c.

2.4.1. The set packing polytope

Feasible solutions of the set packing problem have a nice graph theoretic interpretation. Introduce a node for each column index of A and an edge (i,j) between two nodes i and j if their corresponding columns have a common non-zero entry in some row. The resulting graph, denoted by $G(A)$, is called (*column*) *intersection graph*. Obviously, every feasible 0–1 vector x satisfying $Ax \leqslant \mathbf{1}$ is the incidence vector of a *stable set* ($U \subseteq V$ is a stable set if $i,j \in U$ implies $(i,j) \notin E$) in the graph $G(A)$. Conversely, the incidence vector of any stable set in $G(A)$ is a feasible solution of the set packing problem $Ax \leqslant \mathbf{1}$. So a study of stable sets in graphs is equivalent to a study of the set packing problem.

Now consider some 0–1 matrix A and denote by

$$P(A) = \operatorname{conv}\{x \in \{0,1\}^N\colon Ax \leqslant \mathbf{1}\}$$

the *set packing polytope*. Let $G=(V,E)$ be the intersection graph $G(A)$. From our previous discussion it follows that $P(A)=\operatorname{conv}\{x \in \{0,1\}^n\colon x_i+x_j \leqslant 1, (i,j)\in E\}$, where the latter is an integer programming formulation of the stable set problem in G. In other words, with two matrices A and A' one may associate the same set packing polytope if and only if their corresponding intersection graphs coincide. It is therefore customary to study $P(A)$ via the graph G and denote the set packing polytope and the stable set polytope, respectively, by $P(G)$.

The following observations about $P(G)$ are immediate:

(i) $P(G)$ is full dimensional.
(ii) $P(G)$ is down monotone, i.e., $x \in P(G)$ implies $y \in P(G)$ for all $0 \leqslant y \leqslant x$. All non-trivial facets of $P(G)$ have non-negative coefficients.
(iii) The non-negativity constraints $x_j \geqslant 0$ induce facets of $P(G)$.

It is also well known that the edge and non-negativity constraints suffice to describe $P(G)$ if and only if G is bipartite (i.e., there is a partition (V_1, V_2) of the nodes such that every edge has one endpoint in V_1 and the other in V_2).

Non-bipartite graphs contain odd cycles. Odd cycles give rise to new valid inequalities that cannot be derived as linear combinations of the edge inequalities.

Proposition 2.15 (Padberg [84]). *Let $C \subseteq E$ be a cycle of odd cardinality in G. The* odd cycle inequality

$$\sum_{i\in V(C)} x_i \leqslant \frac{|V(C)|-1}{2}$$

is valid for $P(G)$. It defines a facet of $P((V(C)), E(V(C)))$ *if and only if C is an odd hole, i.e., a cycle without chords.*

Odd cycle inequalities can be separated in polynomial time using the algorithm of Lemma 9.1.11 in [50] based on shortest paths. Graphs $G=(V,E)$ for which $P(G)$ is completely described by the *edge inequalities* $x_i + x_j \leqslant 1$ for $(i,j)\in E$ and the odd cycle inequalities are called *t-perfect*. This notion was introduced in [31] and includes series parallel and bipartite graphs.

Another important class of valid inequalities for the stable set polytope are clique inequalities.

Proposition 2.16 (Fulkerson [44], Padberg [84]). *Let $(C, E(C))$ be a clique in G. The inequality*

$$\sum_{i \in C} x_i \leqslant 1$$

is valid for $P(G)$. It defines a facet of $P(G)$ if and only if $(C, E(C))$ is maximal with respect to node-inclusion.

Graphs $G = (V, E)$ for which $P(G)$ is completely described by the clique inequalities are called *perfect*, a notion going back to Berge [19].

Unlike the class of odd cycle inequalities, the separation problem for the class of clique inequalities is NP-hard, see Theorem 9.2.9 in [50]. Surprisingly, however, there exists a larger class of inequalities, called *orthonormal representation inequalities* (see Proposition 3.5), that includes the clique inequalities and that can be separated in polynomial time. See Section 2.3 for a further discussion. Besides cycle, clique and OR-inequalities, there are many other inequalities known for the stable set polytope. Among these are blossom, odd antihole, wheel, antiweb and web, wedge inequalities and many more. Reference [23] gives a survey on these inequalities including a discussion on their separability.

2.4.2. The independence system polytope

Independence systems provide a framework in combinatorial optimization that generalizes among others the feasible sets of knapsack and set packing problems. To see this, let N be a finite ground set. A collection $\mathscr{I}$ of subsets of N is an *independence system* if it is closed under taking subsets, i.e.,

$$F \in \mathscr{I} \text{ and } G \subseteq F \text{ implies } G \in \mathscr{I}.$$

Associated with an independence system is a second system $\mathscr{C}$ of subsets of N. $\mathscr{C}$ is called the system of *circuits*. It includes all subsets of N of minimal cardinality that do not belong to $\mathscr{I}$.

From the definition of an independence system it is clear that, for instance, the set of all feasible points in a 0–1 knapsack set forms an independence system, and the minimal covers are the circuits. Also the set of stable sets in a graph forms an independence system. Here the cardinality of each circuit is two, and the circuits are precisely the edges of the graph.

More generally, let $A \in \mathbb{R}_+^{m \times n}$ be a non-negative matrix. The set of all 0–1 solutions satisfying $Ax \leqslant b$ for $b \in \mathbb{R}^m$ forms an independence system $\mathscr{I}$ on the ground set $N = \{1, \ldots, n\}$. Let

$$P_{\mathscr{I}} := \text{conv}\{x \in \{0,1\}^n : Ax \leqslant b\}.$$

$P_{\mathscr{I}}$ is called an independence system polyhedron. The following fact about the facet-defining inequalities of $P_{\mathscr{I}}$ is immediate.

Proposition 2.17. *Let $c^{\mathrm{T}}x \leqslant \gamma$ be a facet-defining inequality that is not a positive multiple of one of the non-negativity constraints $-x_i \leqslant 0$. Then c is a non-negative vector and $\gamma > 0$.*

Observe that for the set packing problem Proposition 2.17 was stated in (ii) in Section 2.4.1. An easy example of a valid inequality for the polyhedron of a general independence system is the *circuit constraint*.

Proposition 2.18. *Let $\mathscr{I}$ be an independence system and let $C \subseteq N$ be a circuit. The inequality*

$$\sum_{i \in C} x_i \leqslant |C| - 1$$

is valid for $P_{\mathscr{I}}$.

In fact the problem of finding a maximum weight set in an independence system can be formulated as the integer program

$$\max\left\{c^{\mathrm{T}}x\colon \sum_{i \in C} x_i \leqslant |C| - 1 \text{ for all } C \in \mathscr{C},\ x \in \{0,1\}^N\right\}.$$

Except for special cases, a circuit constraint does not necessarily define a facet of the associated independence system polyhedron. Recall that this applies in particular to the stable set problem for which clique constraints subsume the edge constraints. This motivates the following definition.

Definition 2.19. For $T \subseteq N$, the inequality

$$\sum_{i \in T} x_i \leqslant r(T) := \max\{|S|\colon S \subseteq T, S \in \mathscr{I}\}$$

is called a *rank inequality*, since the right-hand side reflects the maximal cardinality of an independence set with support in T.

Calculating the rank of a set is typically a difficult problem. For instance for the stable set problem, the rank inequality for an arbitrary graph G takes the form

$$\sum_{i \in V} x_i \leqslant \alpha(G),$$

where $\alpha(G)$ is the size of a maximum stable set in G, and it is NP-hard to calculate its value.

If $\mathscr{I}$ is an arbitrary independence system, then one cannot expect to derive a system of inequalities that describes $P_{\mathscr{I}}$. This motivates the search for a partial description. A natural starting point is again the stable set polyhedron. Specifically, we can think of an odd cycle on $\{1,\dots,2k+1\}$ as a set of adjacent pairs $e_i=(i,i+1) \bmod 2k+1$ for $i=1,\dots,2k+1$ such that at most one item can be chosen from each pair.

Generalizing, we now consider a set $\{1,\dots,n\}$ and the set of adjacent t-tuples $N^i=\{i,i+1,\dots,i+t-1\} \bmod n$ for $i=1,\dots,n$. For $q \leqslant t$, the set consisting of all sets containing at most $q-1$ elements from each set N^i is an independence system, known as an *antiweb*, denoted $\mathscr{AW}(n,t,q)$. Thus

$$\mathscr{AW}(n,t,q) := \{I \subseteq N \colon |I \cap N^j| \leqslant q-1 \text{ for all } j=1,\dots,n\}.$$

For example the antiweb $\mathscr{AW}(5,3,3)$ is the set of subsets represented by the feasible incidence vectors of the 0–1 integer program with constraints

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} x \leqslant \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}.$$

The set $\mathscr{C}$ of all circuits of $\mathscr{AW}(n,t,q)$ is equal to

$$\mathscr{C} := \{C \subseteq N \colon |C| = q, C \subseteq N^j \text{ for some } j \in \{1,\dots,n\}\}.$$

An antiweb gives rise to a valid inequality for the associated independence system polyhedron $P_{\mathscr{I}}$. In the example of $\mathscr{AW}(5,3,3)$, the inequality reads $\sum_{i\in N} x_i \leqslant 3$. More generally, one obtains

Proposition 2.20. *Let $\mathscr{AW}(n,t,q)$ be an antiweb and $P_{\mathscr{I}}$ the associated polyhedron. The inequality $\sum_{i\in N} x_i \leqslant \lfloor n(q-1)/t \rfloor$, called an antiweb inequality, is valid for $P_{\mathscr{I}}$.*

Proof. The sum of all constraints $\sum_{i\in N^j} x_i \leqslant q-1$ for $j=0,\dots,n-1$ reads $\sum_{i\in N} tx_i \leqslant n(q-1)$. Therefore, the antiweb inequality coincides with the Chvátal–Gomory cutting plane $\sum_{i\in N} x_i \leqslant \lfloor n(q-1)/t \rfloor$ that is valid for $P_{\mathscr{I}}$. □

No polynomial time algorithms are known for the antiweb inequalities. For an antiweb $\mathscr{AW}(n,t,q)$ the associated inequality always defines a facet if $n=t$. Hence we may assume that $n>t$. In this case a necessary condition for the antiweb inequality to define a facet of $P_{\mathscr{I}}$ is that t is not a divisor of $n(q-1)$. This condition is also sufficient. This condition, the definition of an antiweb and Proposition 2.20 are taken from Laurent [64]. The antiweb inequality in Laurent's paper extends, in particular, the generalized odd holes and antiholes of [42]. It also includes generalized cliques that were introduced in [77]. There are various other families of inequalities known for the independence system that we refrain from discussing here in detail.

Very special independence systems in which the rank inequalities and non-negativity constraints suffice to describe the convex hull $P_{\mathscr{I}}$ include matroids, see [40]. A generalization of the result to the intersection of two matroids can be found in [41].

2.4.3. The set covering polytope

The feasible solutions of the set covering problem

$$\{x \in \{0,1\}^n\colon Ax \geqslant \mathbf{1}\}$$

are in one-to-one correspondence with the independent sets of the system $\mathscr{I}$

$$\left\{ \bar{x} \in \{0,1\}^n\colon \sum_{j \in C} \bar{x}_j \leqslant |C| - 1 \text{ for } C \in \mathscr{C} \right\},$$

when the rows of A correspond to the incidence vectors of circuits $C \in \mathscr{C}$ and $\bar{x}_j = 1 - x_j$ for $j \in N = \{1, \ldots, n\}$.

Note that the antiweb inequality has an equivalent counterpart for the set covering polytope that is derived by complementing every binary variable. In fact the (q,t) roses of [93] are precisely Laurent's antiweb inequalities, see also [80]. Further inequalities for the set covering polytope have been derived, see [23] for a survey, but again all separation algorithms known are of heuristic nature.

3. Extensions

So far we have tried to introduce various ways to derive cutting planes for integer and mixed integer programs of potential computational value. There are many further extensions that are algorithmically promising and worth further exploration. Below we discuss three such topics: the idea of mixing MIR inequalities, the approach of constructing discrete relaxations of integer programs, and the use of semidefinite programming for separation issues.

3.1. Mixing MIR inequalities

Consider the mixed integer set

$$X = \{(x,s) \in \mathbb{Z}^{|P|} \times \mathbb{R}_+\colon s + Cx_i \geqslant b_i, i \in P\}, \text{ for } P = \{1, \ldots, p\}.$$

Let $\mu_i = \lceil b_i/C \rceil$ and $r_i = b_i - (\mu_i - 1)C$. We assume that the constraints defining X are ordered in such a way that $r_i \leqslant r_{i+1}$.

The MIR inequality associated with each constraint $i \in P$ of X is

$$s \geqslant r_i(\mu_i - x_i).$$

By "mixing" these inequalities, a new inequality is obtained.

Proposition 3.1 (Günlük and Pochet [56]). *Taking* $r_0 = 0$, *the inequality*

$$s \geqslant \sum_{i \in P} (r_i - r_{i-1})(\mu_i - x_i)$$

is valid for X.

We illustrate the mixing procedure on two examples.

Example 3.2. Consider an instance of a discrete constant capacity lot-sizing problem,

$$X = \{(x,s) \in \{0,1\}^3 \times \mathbb{R}^4_+ \colon s_{i-1} + Cx_i = b_i + s_i, i \in \{1,2,3\}\},$$

where $C=10$, $b_1=6$, $b_2=7$ and $b_3=8$. Eliminating variables s_1, s_2 and s_3, we obtain the inequalities,

$$\begin{aligned} s_0 + 10x_1 + 10x_2 + 10x_3 &\geqslant 21, \\ s_0 + 10x_1 + 10x_2 &\geqslant 13, \\ s_0 + 10x_1 &\geqslant 6, \end{aligned}$$

to which we can associate the MIR inequalities

$$\begin{aligned} s_0 &\geqslant 3 - x_1 - x_2 - x_3, \\ s_0 &\geqslant 3(2 - x_1 - x_2), \\ s_0 &\geqslant 6(1 - x_1). \end{aligned}$$

Applying Proposition 3.1, we obtain the mixed MIR inequality

$$S_0 \geqslant (3 - x_1 - x_2 - x_3) + 2(2 - x_1 - x_2) + 3(1 - x_1).$$

In [56] it is shown that every (k,l,S,I) inequality for the constant capacity lot-sizing problem can be obtained by mixing MIR inequalities. These inequalities suffice to solve the constant capacity lot-sizing problem by linear programming when the objective function satisfies the Wagner–Whitin assumption [89]. See Section 3.2 for a more extensive discussion of inequalities for lot-sizing problems.

Mixing can also be used to derive valid inequalities for general integer programs.

Example 3.3. Consider the following integer set

$$X = \{x \in \mathbb{Z}^5_+ \colon x_1 + 3x_2 + 10x_4 \geqslant 25, x_1 + 2x_3 + 10x_5 \geqslant 37\}.$$

Defining $s = x_1 + 3x_2 + 2x_3$, the two constraints defining set X can be relaxed to give a set

$$X' = \{(x_4,x_5,s) \in \mathbb{Z}^2_+ \times \mathbb{R}^1_+ \colon s + 10x_4 \geqslant 25, s + 10x_5 \geqslant 37\}.$$

Applying Proposition 3.1 to X', we obtain the mixed MIR inequality

$$s \geqslant 5(3 - x_4) + 2(4 - x_5)$$

or equivalently

$$x_1 + 3x_2 + 2x_3 \geqslant 5(3 - x_4) + 2(4 - x_5)$$

a valid inequality for X.

Other examples of application of the mixing idea can be found in [56].

3.2. Set packing relaxations

In the introduction it was mentioned that knowledge about the set packing polytope can be used to strengthen certain integer programming formulations. Below we show by example how, by introducing additional variables, it is possible to derive a set packing relaxation, generate one or more valid inequalities, and then project back into the original space of variables. We then give a formal description of the approach.

Example 3.4. Let P_I be the convex hull of all 0–1 vectors that satisfy the system of inequalities

$$\begin{aligned} 5x_1 + 5x_2 + 7x_3 + 2x_4 \qquad\qquad &\leqslant 18,\\ 8x_3 + x_4 + 6x_5 + 5x_6 &\leqslant 19,\\ 7x_1 + 2x_2 + \qquad\qquad 7x_5 + x_6 &\leqslant 16. \end{aligned}$$

Define variables $w_1 = x_1x_2$, $w_2 = x_3x_4$ and $w_3 = x_5x_6$, so that $x_1, x_2 \geqslant w_1$, $x_1 + x_2 - 1 \leqslant w_1$, etc. From the first constraint we have that $10w_1 + 9w_2 \leqslant 18$, $w_1, w_2 \in \{0,1\}$ from which we obtain the valid cover inequality $w_1 + w_2 \leqslant 1$. Similarly, from the second and third constraints, we obtain $w_2 + w_3 \leqslant 1$ and $w_1 + w_3 \leqslant 1$. Now the odd cycle (or clique) inequality $w_1 + w_2 + w_3 \leqslant 1$ is valid, leading finally to a valid inequality in the original variables $(x_1 + x_2 - 1) + (x_3 + x_4 - 1) + (x_5 + x_6 - 1) \leqslant 1$ or $\sum_{i=1}^{6} x_i \leqslant 4$ which is valid for P_I.

In general consider a 0–1 integer program $\max\{c^{\mathrm{T}}x\colon Ax \leqslant b,\ x \in \{0,1\}^n\}$ and let $P_I = \operatorname{conv}\{x \in \mathbb{Z}^n\colon Ax \leqslant b, 0 \leqslant x \leqslant \mathbf{1}\}$. We define a set of affine functions $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i = 1, \ldots, M$ with the property that $f_i(x) \leqslant 1$ and $f_i(x) \in \mathbb{Z}$ for $x \in P_I \cap \mathbb{Z}^N$. We define a graph G, called the *conflict graph*, by introducing a node for each of these M affine functions and edges (i, j) if $f_i(x) + f_j(x) \leqslant 1$ for all $x \in P_I$. Now it is readily seen that any valid inequality for the stable set polytope $P(G)$ associated with the conflict graph G yields a valid inequality for P_I.

Natural affine functions that come up are $f_i(x) = x_j$ or $f_i(x) = 1 - x_j$. These are the ones that are generally used in mixed integer programming solvers, see, for instance, [5,36,61]. In the above example we have used the affine functions $f_1(x) = x_1 + x_2 - 1$, $f_2(x) = x_3 + x_4 - 1$, $f_3(x) = x_5 + x_6 - 1$.

More complicated affine functions have been used in Borndörfer and Weismantel [25]. It is shown that various inequalities known for certain combinatorial optimization problems can be interpreted as inequalities from a set packing relaxation. For instance, it turns out that two-chorded cycle inequalities for the clique partitioning problem are odd cycle inequalities of an appropriate set packing relaxation, and that a large class of Möbius ladder inequalities and fence inequalities for the acyclic subdigraph problem are cycle and clique inequalities, respectively, of suitable set packing relaxations.

3.3. Polynomial separation algorithms via matrix cuts

Coming back to our earlier discussions on the stable set polytope, we indicated that there are polynomial time separation algorithms for various classes of valid inequalities, but that a polynomial time separation algorithm cannot be expected for the family of clique constraints. More striking is the fact that clique constraints can be generalized, and that this larger family can be separated in polynomial time. This result is one of the most appealing applications of semidefinite programming in combinatorial optimization, see [50,66].

Let $G=(V,E)$ be a graph with $|V|=n$ and $P(G)$ the associated stable set polyhedron. By P we denote the fractional stable set polytope. For $s\in\mathbb{Z}_+$ a sequence of vectors of unit length, $v^1,\ldots,v^n\in\mathbb{R}^s$, $||v^i||=1$, $i=1,\ldots,n$ is called an *orthonormal representation* of G if $(i,j)\notin E$ implies that $(v^i)^{\mathrm{T}}v^j=0$.

An orthonormal representation of G and a vector of unit length $c\in\mathbb{R}^s, ||c||=1$ lead to a valid inequality for the stable set polyhedron.

Proposition 3.5. *Let $v^1,\ldots,v^n\in\mathbb{R}^s$ be an orthonormal representation of G and $c\in\mathbb{R}^s$, $||c||=1$. The inequality*

$$\sum_{i\in V}(c^{\mathrm{T}}v^i)^2x_i\leqslant 1,$$

called an orthonormal representation (OR)-inequality, is valid for $P(G)$, and every clique inequality is an OR-inequality.

Proof. Let χ^S be the incidence vector of a stable set S in G. Then $(v^i)^{\mathrm{T}}v^j=0$ for all $i,j\in S$, $i\neq j$. We can express c as $c=\sum_{j\in S}\lambda_jv^j+\tilde{c}$ with $\lambda\in\mathbb{R}^S$ and $\tilde{c}$ in the orthogonal complement of the linear space induced by the vectors v^j, $j\in S$. Then

$$\sum_{i\in V}(c^{\mathrm{T}}v^i)^2\chi_i^S=\sum_{i\in S}(c^{\mathrm{T}}v^i)^2=\sum_{i\in S}\lambda_i^2\leqslant 1,$$

because $||c||=1$.

If Q is a clique in G we may set $v^i=c=e^1\in\mathbb{R}^n$ for all $i\in Q$, and $v^j=e^j$ for all $j\notin Q$. The corresponding orthonormal representation constraint is precisely the clique constraint $\sum_{i\in Q}x_i\leqslant 1$. □

In the following we denote

$$TH(G)=\{x\in\mathbb{R}^n_+\colon x \text{ satisfies all OR-constraints}\}.$$

$TH(G)$ is a convex set that is a relaxation of $P(G)$. It is polyhedral if and only if G is perfect, see [50]. However, even when G is not perfect, one can optimize linear functions over $TH(G)$ in polynomial time. This in turn means that we can separate over $TH(G)$ in polynomial time, and thus satisfy all the OR-inequalities.

To get an impression why this is true, we indicate below how $TH(G)$ can be characterized via positive semidefinite matrices. This result is due to Lovász and Schrijver

[66]. Let

$$
\begin{aligned}
H(G) = \{ & Y \in \mathbb{R}^{V\cup\{v_0\}} \times \mathbb{R}^{V\cup\{v_0\}}: \\
& Y \text{ symmetric}, \\
& Y_{ii} = Y_{i0}\ \forall i \in V, \\
& Y_{ij} = 0\ \forall (i,j) \in E, \\
& Y \text{ positive semidefinite}, \\
& e_0^{\mathrm{T}} Y e_0 = 1 \\
\}. &
\end{aligned}
$$

Theorem 3.6.

$$TH(G) = \{Ye^0\colon\ Y \in H(G)\}.$$

It follows, for instance, from the theory of interior point algorithms that, subject to certain conditions, linear functions can be optimized over the cone of symmetric positive semidefinite matrices subject to linear constraints in polynomial time to within a specified error. Since the constraints in Theorem 3.6 are linear in the space of $(n+1)\times(n+1)$ matrices and the conditions are satisfied, this applies to $TH(G)$.

In fact, $TH(G)$ is the projection of a semidefinite relaxation of the stable set problem. Notice that for any incidence vector x of a stable set we have that

$$x_i + x_j \leqslant x_0 \quad \forall (i,j) \in E \text{ with } x_0 = 1.$$

Therefore, the symmetric $(n+1)\times(n+1)$ matrix

$$\tilde{X} = \begin{bmatrix} x_0^2 & x_0 x^{\mathrm{T}} \\ x_0 x & x x^{\mathrm{T}} \end{bmatrix}$$

satisfies the condition that

(a) $\tilde{X}_{ij} = 0$ for all $(i,j) \in E$.
(b) $\tilde{X}_{00} = 1$.
(c) $\tilde{X}_{ii} = \tilde{X}_{i0}$ for all $i \in V$.
(d) $v^{\mathrm{T}}\tilde{X}v \geqslant 0$ for all $v \in \mathbb{R}^{n+1}$, i.e., $\tilde{X}$ is positive semidefinite.

Neglecting the condition that the $n \times n$ submatrix of $\tilde{X}$ is of the form xx^{T}, we end up with a relaxation of the stable set problem in the space of the symmetric $(n+1)\times(n+1)$ matrices. Projecting back to the space of x-variables (using the standard lift-and-project approach) yields precisely $TH(G)$. Important is the fact that $TH(G)$ can be strengthened by using further information in quadratic space about the matrices associated with stable sets and projecting back to the space of x-variables. This follows from the work of Lovász and Schrijver [66] on matrix cuts. We also refer to [50,65]. The conditions to be encountered in the quadratic space come from multiplying each constraint of the fractional stable set problem in the original space by

x_i and by $(1-x_i)$, replacing the quadratic terms by the corresponding matrix variable and requiring that $x_i^2 = x_i$.

Theorem 3.7. *Let*

$$\begin{aligned} T(G) = \{Ye^0 \colon\ & Y \in H(G) \\ & u^{\mathrm{T}} Ye_i \geqslant 0\ \forall u \in \operatorname{cone}(\{1\} \times P)',\ i = 1, \dots, n \\ & u^{\mathrm{T}} Y(e^0 - e_i) \geqslant 0\ \forall u \in \operatorname{cone}(\{1\} \times P)',\ i = 1, \dots, n \\ \}, & \end{aligned}$$

where P denotes the fractional stable set polytope and $\operatorname{cone}(\{1\} \times P)'$ *denotes the polar of the set* $\operatorname{cone}(\{1\} \times P)$. *Then the following is true*:

$$\begin{aligned} T(G) \subseteq \{x \geqslant 0 : & \ x \text{ satisfies all edge constraints}, \\ & \ x \text{ satisfies all OR-constraints}, \\ & \ x \text{ satisfies all odd hole constraints}, \\ & \ x \text{ satisfies all odd antihole constraints}, \\ & \ x \text{ satisfies all odd wheel constraints} \\ \}. & \end{aligned}$$

4. Valid inequalities for some structured MIPs

Here, we look briefly at four problem areas that provide a large variety of applications: fixed charge network design, production planning, facility location and electricity generator scheduling. As many of the ideas for generating inequalities for network problems can be used in the other areas, we start with network design.

4.1. Fixed charge network design

Traditionally, single commodity fixed charge network problems arose in designing transport, water and electricity networks. In the last 10 years the design of telecommunication networks and VLSI have provided perhaps the bulk of applications in this area—these include both single commodity problems, such as the construction of two or multiply connected networks, and multicommodity problems which arise because messages/communications between two nodes A and B are distinct from messages being sent from C to D.

Below, we concentrate mainly on single commodity problems because the majority of valid inequalities can be explained in this simpler context. We present a variety of different ways to derive inequalities. In particular, we first look at the simplest single node model considering different variants, uncapacitated and capacitated, and with 0–1 or integer variables as appropriate. The same single node inequalities are then used when several nodes S are combined to form a macro-node, but the difficulty is now how to choose the set S.

We then present four classes of inequalities that use more of the network structure, such as the sparsity of the network, ways to combine different dicut inequalities, or submodularity.

Finally, we briefly touch on multicommodity problems with a single source and sink for each commodity. We look at a basic single arc model with both divisible and indivisible flows, and then again at how to choose a good macro-node set S on which to generate a dicut or other inequality.

4.1.1. Single commodity problems

We consider a basic single commodity fixed charge network flow problem consisting of a digraph $D=(V,A)$ and a vector $b\in\mathbb{R}^n$ with $\sum_{i=1}^n b_i=0$, where $n=|V|$, $T=\{i\in V\colon b_i>0\}$ is the set of demand nodes or *terminals*, and $U=\{i\in V\colon b_i<0\}$ is the set of *sources*. Given unit flow costs p_{ij} and fixed arc capacity installation costs f_{ij} for an amount C_{ij} of capacity on arc $(i,j)\in A$, the problem is to find a feasible flow minimizing the sum of the flow and capacity installation costs. Much of the literature has been devoted to the special case of this problem without flow costs—special cases are the Steiner tree problem, or the problem of designing a two-connected network of minimum cost, etc. [48,54].

Below we use the notation $\bar{S}=V\setminus S$, $V^-(i)=\{j\in V\colon (j,i)\in A\}$, $V^+(i)=\{j\in V\colon (i,j)\in A\}$, and $\delta(S,\bar{S})=\{(i,j)\in A\colon i\in S, j\in\bar{S}\}$.

Letting y_{ij} denote the flow in arc $(i,j)\in A$ and x_{ij} the number of times the capacity C_{ij} is installed, we obtain the natural formulation

$$\min \sum_{(i,j)\in A} p_{ij}y_{ij} + \sum_{(i,j)\in A} f_{ij}x_{ij}, \tag{25}$$

$$\sum_{j\in V^-(i)} y_{ji} - \sum_{j\in V^+(i)} y_{ij} = b_i \quad \text{for } i\in V, \tag{26}$$

$$0\leqslant y_{ij}\leqslant C_{ij}x_{ij} \quad \text{for } (i,j)\in A, \tag{27}$$

$$x_{ij}\in\mathbb{Z}^1_+ \quad \text{for } (i,j)\in A. \tag{28}$$

Here (26) are flow conservation constraints and (27) are variable upper bound capacity constraints. We will denote the feasible region (26)–(28) by X^{FC}. In practice one also encounters many variants such as

(i) $x_{ij}\in\{0,1\}$ in place of (28),
(ii) $C_{ij}=C$ and also possibly $f_{ij}=f$ for all $(i,j)\in A$ when standard equipment is installed throughout the network,
(iii) Capacity C^0_{ij} already exists on certain arcs, and two or more different types of capacity can be installed, so we have $0\leqslant y_{ij}\leqslant C^0_{ij}+C^1x^1_{ij}+C^2x^2_{ij}$ in place of (27),
(iv) Capacity is undirected, so we have $0\leqslant y_{ij}+y_{ji}\leqslant C_ex_e$ in place of (27), where e represents the edge (i,j).

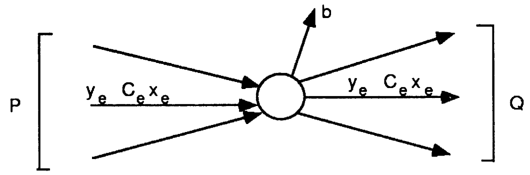

Fig. 2. Single node flow set.

4.1.2. Single node inequalities

If we just consider the flow conservation constraint (26) for node i along with the associated bounds on the flows (27), we obtain the situation shown in Fig. 2 and the corresponding single node flow set

$$X^{SN} = \left\{ (x,y) \in \mathbb{Z}_+^{p+q} \times \mathbb{R}_+^{p+q} \colon \sum_{e\in P} y_e - \sum_{e\in Q} y_e = b,\ y_e \leqslant C_e x_e \text{ for } e \in P \cup Q \right\}$$

with $p=|P|$ and $q=|Q|$, and its relaxation

$$X_>^{SN} = \left\{ (x,y) \in \mathbb{Z}_+^{p} \times \mathbb{R}_+^{p} \colon \sum_{e\in P} y_e \geqslant b,\ y_e \leqslant C_e x_e \text{ for } e \in P \right\}.$$

The uncapacitated case. If the capacities are so large that the flow on each arc is unrestricted, x_e can be restricted to be a 0–1 variable for all arcs $e \in A$. Now points in $X_>^{SN}$ satisfy $\sum_{e\in P} y_e \geqslant b$, $C\sum_{e\in P} x_e \geqslant b$, and thus if $b>0$, the *cut* inequality

$$\sum_{e\in P} x_e \geqslant 1$$

is valid for $X_>^{SN}$. Note that if $b<0$, a similar inequality is obtained with Q in place of P.

More generally, if F is a subset of the arcs in P, feasible points in $X_>^{SN}$ satisfy $\sum_{e\in P\setminus F} y_e + C\sum_{e\in F} x_e \geqslant b$ leading to the *mixed cut* inequality

$$\sum_{e\in P\setminus F} y_e + b\sum_{e\in F} x_e \geqslant b.$$

The constant capacity case—integer batches. For simplicity, we assume that the capacities C_e and demands b are integer. When $b>0$ and $C_e = C$ for all $e\in F$, the inequality $\sum_{e\in P\setminus F} y_e + C\sum_{e\in F} x_e \geqslant b$ leads to the *residual capacity* or MIR inequality (see Section 1.2) for $X_>^{SN}$

$$\sum_{e\in P\setminus F} y_e + r\sum_{e\in F} x_e \geqslant r\mu, \tag{29}$$

where $\mu = \lceil b/C \rceil$ and $r = b - (\mu-1)C$.

For X^{SN}, with $G \subseteq Q$, the inequality takes the more general form

$$\sum_{e\in P\setminus F} y_e + r\sum_{e\in F} x_e \geqslant r\mu + \sum_{e\in G} [y_e - (C-r)x_e], \tag{30}$$

see [4].

The capacitated 0–1 *case*. Rewriting the simple flow cover inequalities for single node flow sets that have been described in Section 1.1, we first present valid inequalities for $X^{SN}_{>} \cap \{(x,y) \in \{0,1\}^p \times \mathbb{R}^p_+\}$. For F a cover, ($\sum_{e\in F} C_e - b = \lambda > 0$), we obtain

$$\sum_{e\in P\setminus F} y_e + \sum_{e\in F} (C_e - \lambda)^+ x_e \geqslant \sum_{e\in F} (C_e - \lambda)^+.$$

Generalizing to include outflows, the basic inequality obtained for $X^{SN} \cap \{(x,y) \in \{0,1\}^{p+q} \times \mathbb{R}^{p+q}_+\}$ is

$$\sum_{e\in P\setminus F_1} y_e + \sum_{e\in F_1} (C_e - \lambda)^+ x_e \geqslant \sum_{e\in F_1} (C_e - \lambda)^+ + \sum_{e\in F_2} (y_e - C_e) + \sum_{e\in L_2} (y_e - \lambda x_e),$$

where $F_1 \subseteq P, F_2, L_2 \subseteq Q, F_2 \cap L_2 = \emptyset$ and $\sum_{e\in F_1} C_e - \sum_{e\in F_2} C_e - b = \lambda > 0$.

In the constant capacity case, the inequalities for the 0–1 case take the same form as (29) and (30), and are known to describe the convex hull of solutions, see [86,4].

More general capacity constraints. Suppose that the constraints

$$y_e \leqslant C_e^0 + C_e^1 x_e^1 + C_e^2 x_e^2$$

describe the potential capacities. Feasible points now satisfy $\sum_{e\in P\setminus F} y_e + C^1 \sum_{e\in F} x_e^1 + C^2 \sum_{e\in F} x_e^2 \geqslant b - \sum_{e\in F} C_e^0$. Now assuming $b - \sum_{e\in F} C_e^0 > 0$, and divisible capacities (i.e., C^1 divides C^2), which is often the case in telecommunications applications, extensions of the residual capacity inequalities have been proposed in [20,67], and these have been generalized to handle an arbitrary number of divisible capacities in [90].

4.1.3. Aggregate node inequalities

By summing the flow conservation constraints (26) for $i \in S$, we obtain the set X^S:

$$\sum_{e\in\delta(\bar{S},S)} y_e - \sum_{e\in\delta(S,\bar{S})} y_e = \sum_{i\in S} b_i, \tag{31}$$

$$0 \leqslant y_e \leqslant C_e x_e, \quad x_e \in \{0,1\} \text{ for } e \in \delta(\bar{S},S) \cup \delta(S,\bar{S}) \tag{32}$$

which is precisely in the form of the single node flow set X^{SN}. Thus, if $\sum_{i\in S} b_i > 0$, all the inequalities presented above can be generalized to the set X^S. In particular in the uncapacitated case we obtain the *dicut* inequality

$$\sum_{e\in\delta(\bar{S},S)} x_e \geqslant 1$$

and if F is a subset of $\delta(\bar{S},S)$, the *mixed dicut* inequality

$$\sum_{e\in\delta(\bar{S},S)\setminus F} y_e + \left(\sum_{i\in S} b_i\right) \sum_{e\in F} x_e \geqslant \sum_{i\in S} b_i.$$

There is now however a major question to be answered before we can make use of these inequalities. How should the set S of nodes be chosen, given the huge number of possibilities?

The separation problem for dicut inequalities. Formally, we wish to solve the problem: given a solution (x^*, y^*) satisfying the linear programming relaxation of (25)–(28), does there exist a non-empty subset $S \subset V$ with $\sum_{i\in S} b_i > 0$ and $\sum_{e\in\delta(\bar{S},S)} x_e^* < 1$?

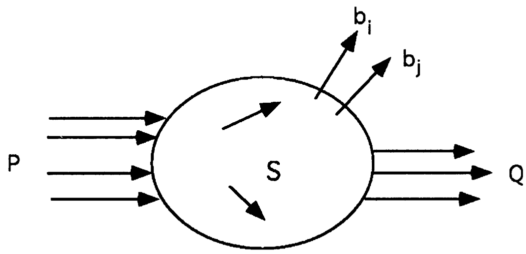

Fig. 3. Aggregate node set.

Special dicut inequalities: *maximum flow*. First we restrict the choice of subsets S. Remember the notation that $T = \{i: b_i > 0\}$ and $U = \{i: b_i < 0\}$. Let $\mathscr{S} = \{S \subset V: S \cap U = \emptyset,\ S \cap T \neq \emptyset\}$. Now if $S \in \mathscr{S}$, we are sure that $\sum_{i \in S} b_i > 0$. The separation problem then reduces to $|T|$ maximum flow problems.

Specifically, choose $s \in U$ and $t \in T$. Let ζ_t be the value of a maximum s–t flow in the digraph $D = (V, A)$ with capacities $h_{ij} = \infty$ if $i, j \in U$ and $h_{ij} = x^*_{ij}$ otherwise. If $\zeta_t \geqslant 1$, there is no violated dicut inequality with $s \in S$ and $t \in T$. Otherwise if $\zeta_t < 1$, the resulting minimal s–t cut gives a violated dicut inequality.

Note that for single source problems with $|U| = 1$, all dicuts of interest are included in this procedure.

All dicut inequalities: *quadratic* 0–1 *Knapsack*. To model the general case, let $z_j = 1$ if $j \in S$ and $z_j = 0$ otherwise. The resulting separation problem can now be written as

$$\zeta = \min \sum_{(i,j) \in A} x^*_{ij}(1 - z_i)z_j,$$

$$\sum_{j \in V} b_j z_j > 0,$$

$$z_j \in \{0, 1\} \quad \text{for } j \in V.$$

If S is the set minimizing ζ, a violated dicut inequality has been found if $\zeta < 1$, and in any case we can look at the single node flow set associated with S for other violated inequalities.

4.1.4. Inequalities using structure

Uncapacitated: *inflow–outflow inequalities*. When all arcs are present in an uncapacitated network, flow entering the network can reach any other node. However, when the network is sparse, this is no longer true. Specifically, consider the subgraph induced by the node set S as shown in Fig. 3. We will now take into account the internal structure of $D_S = (S, A_S)$. Write $P = \delta(\bar{S}, S)$ and $Q = \delta(S, \bar{S})$. Also let $R \subseteq A_S$ be a subset of the arcs in S. For an entering arc $e \in P$, let $S_e = \{i \in S: b_i > 0$ and there exists a dipath in $D_{S,R} = (S, A_S \setminus R)$ from the head of arc e to node $i\}$ and $\alpha_e = \sum_{i \in S_e} b_i$.

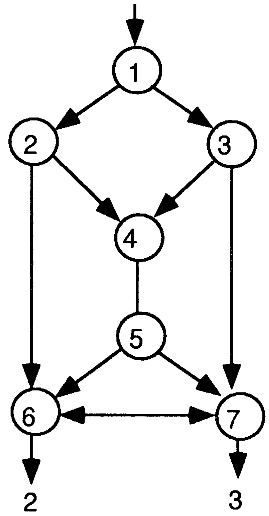

Fig. 4. Network for multicut inequality.

The *inflow–outflow inequality*

$$\sum_{e\in F}\alpha_e x_e + \sum_{e\in(P\setminus F)\cup R} y_e \geqslant \sum_{i\in S} b_i$$

is valid for any $F\subseteq P$.

Uncapacitated: *multi-dicut inequalities*. Rather than use just a single dicut inequality, here we show how to use several dicuts simultaneously. Suppose that for each $t\in T$, a family of dicuts $\{\delta(\bar{S}_t^k, S_t^k)\}_{k=1}^{K_t}$ is given with $t\in S_t^k$ and $S_t^k\cap U=\emptyset$ for all k and t. Also take $F_t^k\subseteq\delta(\bar{S}_t^k, S_t^k)$. The following *multi-dicut inequality*:

$$\sum_{e\in A}\max_{t\in T}\alpha_e(t)y_e + \sum_{e\in A}\sum_{t\in T}\beta_e(t)d_t x_e \geqslant \sum_{t\in T}K_t d_t$$

is shown to be valid in [91], where, for $e\in A$,

$\alpha_e(t)$ is the number of arc sets $\{F_t^k\}_{k=1}^{K_t}$ containing e, and

$\beta_e(t)$ is the number of arc sets $\{\delta(\bar{S}_t^k, S_t^k)\setminus F_t^k\}_{k=1}^{K_t}$ containing e.

Example 4.1. Consider the network shown in Fig. 4 with $T=\{6,7\}$, $d_6=2$ and $d_7=3$. Taking $K_6=K_7=2$, $S_6^1=\{2567\}$, $S_6^2=S_7^1=\{3567\}$, $S_7^2=\{567\}$, $F_6^1=\{(37)\}$, $F_6^2=F_7^1=\{(26)\}$, $F_7^2=\{(26),(37)\}$, we have $\alpha_{26}(1)=\alpha_{37}(1)=\alpha_{37}(2)=1$, $\alpha_{26}(2)=2$, and we obtain the multi-dicut inequality

$$y_{37}+2y_{26}+2x_{12}+10x_{45}+5x_{13}\geqslant 10.$$

0–1 *Capacitated*: *submodular inequalities*. An important, but rare structural property, in discrete optimization problems, is submodularity, which is some discrete form of non-increasing returns. Specifically, $f:\mathscr{P}(N)\to R$ is *submodular* if $f(A)+f(B)\geqslant$

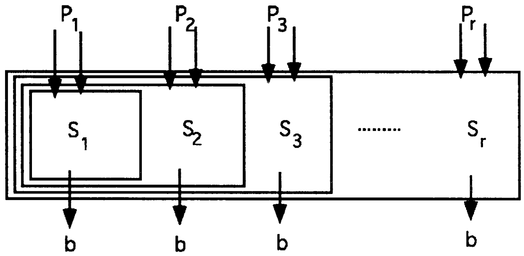

Fig. 5. Embedded node sets.

$f(A\cap B)+f(A\cup B)$ for all $A,B\subseteq N$. Not surprisingly, this structure is reflected in a family of valid inequalities. Consider again Fig. 3. For $F\subseteq P$, let $v(F)$ be the maximum flow that can enter D_S through the arcs of F, and leave via the demand nodes in S with $b_i>0$. It can be shown that v is submodular. Define $\rho_j(T)=v(T\cup\{j\})-v(T)$, and let $\{1,2,\ldots,p\}$ be a chosen ordering of the elements of P. The following *submodular* inequality:

$$\sum_{j\in P} y_j \leqslant v(F)+\sum_{j\in P\setminus F}\rho_j(F\cup\{j+1,\ldots,p\})x_j$$
$$-\sum_{j\in F}\rho_j(F\cap\{j+1,\ldots,p\}\cup\{1,\ldots,j-1\})(1-x_j)+\sum_{e\in Q} y_e$$

is valid, see [109].

Capacitated: *dynamic inequalities*. Here, we use the idea of mixing to combine cut inequalities from different aggregate node sets, which can be viewed as generalizing the use of sparsity in the input–output inequalities. Suppose we have node sets $S_1\subset S_2\subset\cdots\subset S_t$, and entering arcs $P_1,\ldots,P_t$ as shown in Fig. 5. Let $Q_{pq}=\{(i,j)\in A\colon i\in S_p, j\in S_q\}$. Considering the sets $S_1,S_2,\ldots,S_t$ in turn, the inequalities based on the inflow to S_k being at least equal to the demand give

$$\sum_{i=1}^{k}\sum_{e\in P_i} C_e x_e+\sum_{p,q:p>k\geqslant q}\;\sum_{e\in Q_{pq}} y_e\geqslant\sum_{i\in S_k} b_i \tag{33}$$

for $k=1,\ldots,t$.

With constant capacities, the mixing theorem can be applied to give inequalities of the form

$$\sum_{p,q:p>q}\;\sum_{e\in Q_{pq}} y_e\geqslant r_{[1]}(\mu_{[1]}-X_{[1]})+\cdots+(r_{[t]}-r_{[t-1]})(\mu_{[t]}-X_{[t]}),$$

where $\mu_k=\lceil\sum_{i\in S_k} b_i/C\rceil$, $r_k=\sum_{i\in S_k} b_i-(\mu_k-1)C$, $\{[1],\ldots,[t]\}$ is a permutation of $\{1,\ldots,t\}$ with $r_{[1]}\leqslant\cdots\leqslant r_{[t]}$, and $X_{[k]}=\sum_{i=1}^{[k]}\sum_{e\in P_i} x_e$.

Examples of such inequalities are given below both for lot-sizing and for facility location problems.

4.1.5. Multicommodity problems

In multicommodity problems feasible flows have to be determined for each of $k = 1,\dots,K$ commodities satisfying demands b_i^k at each node $i \in V$, where the commodities share arc capacity. This can be formulated as

$$\min \sum_{(i,j)\in A} \sum_k p^k y^k_{(ij)} + \sum_{(i,j)\in A} \sum_k f^k x^k_{(ij)}, \tag{34}$$

$$Ny^k = b^k \quad \text{for } k = 1,\dots,K, \tag{35}$$

$$0 \leqslant \sum_k y^k_{ij} \leqslant C_{ij} x_{ij} \quad \text{for } (i,j) \in A, \tag{36}$$

$$x_{ij} \in Z^1 \quad \text{for } (i,j) \in A, \tag{37}$$

where N is the node-arc incidence matrix of D. In many instances each commodity k has a single source i^k and a single sink j^k, in which case we write $b^k_{j_k} = d_k$, $b^k_{i_k} = -d_k$ and $b^k_i = 0$ otherwise. From now on we limit our attention to this case. We also consider the network loading problem in which x_{ij} is integer rather than 0–1.

4.1.6. Single arc inequalities

Multiple routes: Consider flow in a single arc $(i,j) \in A$. Let y_k be the flow of commodity k in this arc, and x the associated capacity variable. The resulting set is

$$X^{SA} = \left\{ (x,y) \in \mathbb{Z}^1_+ \times \mathbb{R}^K_+ \colon \sum_{k=1}^K y_k \leqslant Cx,\, y_k \leqslant d_k \text{ for } k = 1,\dots,K \right\}.$$

Taking an arbitrary set $K' \subseteq \{1,\dots,K\}$ of commodities and setting $w = \sum_{k\in K'} y_k$, we have that $w \leqslant Cx$ and $w \leqslant \sum_{k\in K'} d_k$ leading to the *arc residual capacity inequality*

$$\sum_{k\in K'} y_k \leqslant \sum_{k\in K'} d_k - r'(\mu' - x),$$

where $\mu' = \lceil \sum_{k\in K'} d_k / C \rceil$ and $r' = \sum_{k\in K'} d_k - (\mu' - 1)C$. It is shown in [67] that this family of inequalities completely describes the convex hull of X^{SA}.

Mono-routing. When each commodity must flow on a single path, the flow of commodity k in arc (i,j) is either 0 or d_k, and so we obtain the knapsack set

$$X^{\mathrm{SAM}} = \left\{ (x_0, x) \in \mathbb{Z}^1_+ \times \{0,1\}^K \colon \sum_k d_k x_k \leqslant Cx_0 \right\}.$$

Valid inequalities for a more general model with capacities of the form $x_1 + Cx_2$ have been derived in [26]. See also [98].

4.1.7. Multinode inequalities

If we choose a commodity k and a set $S \subset V$ with $i^k \in \bar{S}$ and $j^k \in S$, flow conservation for commodity k gives

$$\sum_{e\in\delta(\bar{S},S)} y^k_e - \sum_{e\in\delta(S,\bar{S})} y^k_e = d_k.$$

One can first check for a violated dicut inequality by finding a maximum (i^k, j^k) flow with capacities $\min\{d_k, C_e\} x^*_e$ on the arcs.

More generally, with a constant capacity C and a subset K' of commodities, we have that

$$\sum_{k\in K'}\sum_{e\in\delta(\bar{S},S)} y_e^k \geqslant \sum_{k\in K': i^k\notin S, j^k\in S} d_k,$$

which after introduction of the capacity constraints gives

$$\sum_{e\in\delta(\bar{S},S)} x_e \geqslant \frac{\sum_{k:i^k\notin S, j^k\in S} d_k}{C}$$

and then applying Gomory integer rounding gives

$$\sum_{e\in\delta(\bar{S},S)} x_e \geqslant \left\lceil \frac{\sum_{k:i^k\notin S, j^k\in S} d_k}{C} \right\rceil .$$

Consider now the relaxed version of these inequalities without the round up of the right-hand side term and with $C=1$. They are automatically satisfied by a point x^* if there exists a y such that (x^*, y) satisfies the linear programming relaxation of (34)–(37). More precisely such points satisfy the metric inequalities

$$\sum_e \mu_e x_e \geqslant \sum_k \pi_k d_k,$$

where $\mu \in \mathbb{R}_+^{|E|}$ are arbitrary edge lengths, and π_k is the corresponding length of a shortest path from i^k to j^k, see [59,82]. Note that if $\mu_e = 1$ for $e \in \delta(\bar{S}, S)$, the relaxed inequality above is obtained as a special case.

However, separation for the special case is a max dicut problem, which is NP-hard. Specifically, it suffices to put a weight $-y_e^*$ on each arc of D, and a weight d_k/C on the arcs (i^k, j^k) for $k = 1, \dots, K$, and find a maximum dicut. This separation procedure has been used in [15] in a model with edge capacities and no variable flow costs.

4.2. Lot-sizing

A single-item lot-sizing problem is a very special case of a fixed charge network flow problem, see Fig. 6.

The basic single-item lot-sizing problem is typically formulated as

$$\min \sum_t p_t y_t + \sum_t h_t s_t + \sum_t f_t x_t, \tag{38}$$

$$s_{t-1} + y_t = d_t + s_t \quad \text{for } t = 1, \dots, n, \tag{39}$$

$$y_t \leqslant C_t x_t \quad \text{for } t = 1, \dots, n, \tag{40}$$

$$s_t, y_t \geqslant 0,\ x_t \in \{0, 1\} \quad \text{for } t = 1, \dots, n. \tag{41}$$

Here d_t is the demand, p_t, h_t, f_t are the variable production, storage and fixed setup costs, and C_t is the maximum amount that can be produced in period t. y_t, s_t are continuous variables denoting the production and end-stock in period t, and x_t is a

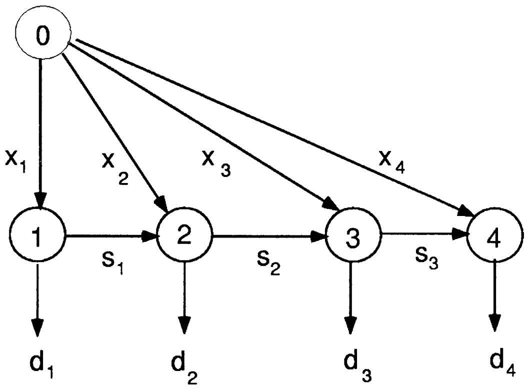

Fig. 6. Network for lot-sizing.

0–1 setup variable indicating whether the machine can produce in period t. Thus $y_t > 0$ only if $x_t = 1$. Constraints (39) are flow balance constraints, and (40) are capacity constraints linking the production and setup variables.

Much is known about the polyhedral structure of different variants of this problem. We will see below that all the valid inequalities can be derived using procedures that we have seen earlier either for general 0–1 MIPs in Sections 1.2 and 2.1, or for fixed charge network problems. Later in this section we will also introduce a natural way to derive valid inequalities for problems with start-ups. Let $d_{kt} = \sum_{j=k}^{t} d_j$.

Uncapacitated lot-sizing. Let X^{ULS} denote the set of feasible solutions of (39)–(41), where again we assume that C is very large and does not limit the amount produced in any period. Aggregating the flow balance constraints (39) for $t = k, \dots, l$, and choosing a subset $S \subseteq \{k, \dots, l\}$, of periods, leads to the relaxation

$$s_{k-1} + \sum_{j \notin S, k \leqslant j \leqslant t} y_j + C \sum_{j \in S, j \leqslant t} x_j \geqslant d_{kt} \tag{42}$$

leading to the MIR inequalities

$$s_{k-1} + \sum_{j \notin S, k \leqslant j \leqslant t} y_j \geqslant d_{kt} \left(1 - \sum_{j \in S, j \leqslant t} x_j \right) \tag{43}$$

and by the mixing procedure (Section 2.1) to the valid inequalities

$$s_{k-1} + \sum_{j \notin S, k \leqslant j \leqslant t} y_j \geqslant \sum_{j \in S} d_j \left(1 - \sum_{t \in S, k \leqslant t \leqslant j} x_t \right). \tag{44}$$

These inequalities completely describe the convex hull of X^{ULS} [16].

Constant capacity lot-sizing. An identical approach leads to a large number of facet-defining inequalities when $C_t = C$ for $t = 1, \dots, n$. First from (42) we obtain the MIR inequality

$$s_{k-1} + \sum_{j \notin S, k \leqslant j \leqslant t} y_j \geqslant r_{kt} \left(\mu_{kt} - \sum_{j \in S, j \leqslant t} x_j \right),$$

where $\mu_{kt} = \lceil d_{kt}/C \rceil$ and $r_{kt} = d_{kt} - (\mu_{kt} - 1)C$.

Now if the r_{kt} are placed in non-decreasing order, and written $r_{[1]} \leqslant r_{[2]} \cdots \leqslant r_{[q]}$, and $\mu_{[i]}$ and $X^S_{[i]}$ are the corresponding terms for μ and $\sum_j x_j$, the mixing procedure gives

$$s_{k-1} + \sum_{j \notin S, k \leqslant j \leqslant l} y_j \geqslant r_{[1]}(\mu_{[1]} - X^S_{[1]}) + (r_{[2]} - r_{[1]})(\mu_{[2]} - X^S_{[2]}) + \cdots + (r_{[q]} - r_{[q-1]})(\mu_{[q]} - X^S_{[q]}).$$

An example of this inequality has been shown in Example 3.2.

Varying capacity lot-sizing. Inequality (42) with varying capacities gives, setting $s' = s_{k-1} + \sum_{j \notin S, k \leqslant j \leqslant t} y_j$, the relaxation

$$s' + \sum_{j \in S, j \leqslant t} C_t x_j \geqslant d_{kt},\ s' \geqslant 0, x_j \in \{0,1\} \quad \text{for } j \in S,$$

for which mixed knapsack inequalities can be generated, see Section 1.1. Alternatively, aggregation of the flow balance constraints gives the inequality $\sum_{j=l}^{l} y_j \leqslant d_{kt} + s_l$, the bounds give us $y_j \leqslant C_j x_j$, and the uncapacitated inequality (43) gives $y_j \leqslant d_{jl} x_j + s_l$. Setting $s_l = 0$ temporarily, we have a single node flow set:

$$\left\{ (x,y) \in \{0,1\}^{k-l+1} \times \mathbb{R}^{k-l+1}_+ : \sum_{j=k}^{l} y_j \leqslant d_{kl}, y_j \leqslant \min[C_j, d_{jl}] x_j \text{ for } j=k,\ldots,l \right\}.$$

Now it suffices to add the term $(+s_l)$ to the right-hand side of any flow cover inequality to have a valid inequality for X^{ULS}, see [91].

4.2.1. Modelling start-ups

If $x_1, x_2, \ldots, x_n \in Z^n_+$ denote the number of machines set-up in periods $1, \ldots, n$, it is often important to know the number $\max[x_t - x_{t-1}, 0]$ of machines that start-up in period t. If z_t is a variable representing the number of start-ups, we use the constraints

$$z_t \geqslant x_t - x_{t-1}, \quad z_t \geqslant 0$$

to get an upper bound on the number of start-ups, and

$$z_t \leqslant x_t, \quad z_t \leqslant u_t - x_{t-1}$$

to try to make the upper bound tight, where u_t is an upper bound on x_t. This provides an exact formulation if $x_t, x_{t-1} \in \{0,1\}$, but it is not tight otherwise.

Observation 4.2. *Let* $\chi_{kt} = \max\{x_k, \ldots, x_t\}$, *then* $x_k + z_{k+1} + \cdots + z_t \geqslant \chi_{kt}$.

Lot-sizing with start-ups. Let z_t be defined as above to take value 1 if and only if $x_t = 1$ and 0, and let χ_{kl} denote the maximum of $(x_k, \ldots, x_l)$. The uncapacitated inequality (44) says essentially that the stock at the end of period $k-1$ contains the demand d_t if there is no production in periods $k,\dots,t$, or in other words if $\chi_{kt}=0$. This gives the valid inequality $s_{k-1}\geqslant\sum_{t=k}^{l}d_t(1-\chi_{kt})$, or using Observation 4.2

$$s_{k-1}\geqslant\sum_{t=k}^{l}d_t(1-x_k-z_{k+1}-\cdots-z_t).$$

In the constant capacity case, either $\chi_{j,l}=0$ and

$$s_{k-1}+C\left(\sum_{i=k}^{j-1}x_i+\chi_{j,l}\right)\geqslant d_{kl}=d_{k,j-1}+d_{jl},$$

or $\chi_{j,l}=1$ and so

$$s_{k-1}+C\left(\sum_{i=k}^{j-1}x_i+\chi_{j,l}\right)\geqslant d_{k,j-1}+C.$$

Thus, all feasible solutions to (42) satisfy $s_{k-1}+C(\sum_{i=k}^{j-1}x_i+\chi_{j,l})\geqslant d_{k,j-1}+\min[C,d_{jl}]$, and from this we obtain the valid MIR inequality

$$s_{k-1}\geqslant\tilde{r}_{kl}\left(\tilde{\mu}_{kl}-\sum_{i=k}^{j-1}x_i-x_j-z_{j+1}-\cdots-z_l\right),$$

where $\tilde{d}_{kl}=d_{k,j-1}+\min[C,d_{jl}]$, $\tilde{\mu}_{kl}=\lceil\tilde{d}_{kl}/C\rceil$ and $\tilde{r}_{kl}=\tilde{d}_{kl}-(\tilde{\mu}_{kl}-1)C$. Now varying l and using mixing, one can obtain the *left extended klSI inequalities* from [32].

4.3. Facility location problems

The capacitated facility location problem is also a special case of the fixed charge network flow problem. One particularity is that the fixed costs are incurred on opening nodes (locations) rather than arcs. We show that both flow cover and dynamic inequalities can be specialized for the special structure of this problem. A more combinatorial class of inequalities, a generalization of inequalities from the uncapacitated case, is also presented.

The feasible region is typically described as follows:

$$\sum_{j\in N}y_{ij}=a_i\quad\text{for } i\in M,\tag{45}$$

$$\sum_{i\in M}y_{ij}\leqslant C_jx_j\quad\text{for } j\in N,\tag{46}$$

$$0\leqslant y_{ij}\leqslant\min[a_i,C_j]x_j\quad\text{for } i\in M,\ j\in N,\tag{47}$$

$$x_j\in\{0,1\}\quad\text{for } j\in N,\tag{48}$$

where y_{ij} is the amount shipped from location j to client i, and $x_j=1$ indicates that location j is in use.

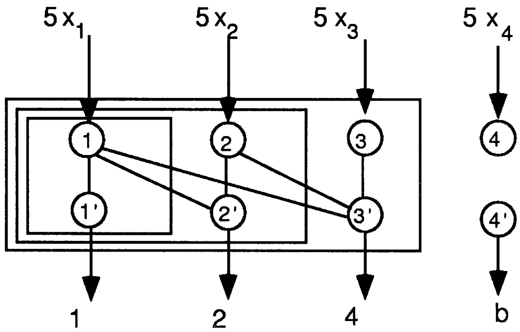

Fig. 7. Dynamic location set.

Letting $v_j = \Sigma_{i\in M} y_{ij}$ and summing up all the demand constraints (45) leads to a single node flow set X described by

$$\left\{ (v,x) \in \mathbb{R}^n_+ \times \{0,1\}^n: \sum_{j\in N} v_j = \sum_{i\in M} a_i,\ 0 \leqslant v_j \leqslant C_j x_j \text{ for } j \in N \right\}$$

for which knapsack and flow cover inequalities can be generated.

Next we consider the internal structure of the underlying digraph. Consider a subset $K \subseteq M$ of clients, a subset $J \subseteq N$ of locations, and for each $j \in J$ a possibly smaller subset $K_j \subseteq K$ of clients. Restricted to this subset, the *effective capacity* of location j is $\bar{C}_j = \min[C_j, \sum_{i\in K_j} a_i]$. Now we obtain a modified flow cover set based on the new variable $\tilde{v}_j = \sum_{i\in K_j} y_{ij}$, namely the set

$$X^{\mathrm{EC}} := \left\{ (\tilde{v},x) \in \mathbb{R}^{|J|}_+ \times \{0,1\}^{|J|}: \sum_{j\in J} \tilde{v}_j \leqslant \sum_{i\in K} a_i, \tilde{v}_j \leqslant \bar{C}_j x_j \text{ for } j \in J \right\}.$$

Specifically, if J is a cover with excess $\lambda = \sum_{j\in J} \bar{C}_j - \sum_{i\in K} a_i > 0$, then we obtain the *effective capacity flow cover inequality*

$$\sum_{j\in J} \sum_{i\in K_j} y_{ij} + \sum_{j\in J} (\bar{C}_j - \lambda)^+(1 - x_j) \leqslant \sum_{i\in K} a_i.$$

Submodular inequalities can also be defined for this model leading to very similar inequalities. The separation problem for the effective capacity and submodular inequalities involves a choice of the sets, J, K and K_j, and is necessarily heuristic, see [1].

Dynamic inequalities. When $K_r \subseteq K_{r-1} \cdots \subseteq K_1$, we can use the embedded set structure to obtain dynamic inequalities, see Section 3.1.

Example 4.3. Consider a problem with four clients and four locations as shown in Fig. 7.

Specifically, we have $J=\{1,2,3\}, K_1=\{1',2',3'\}, K_2=\{2',3'\}$ and $K_3=\{3'\}$. This corresponds to an embedded node set with $S_1 = \{1,1'\}, S_2 = \{1,2,1',2'\}$,

$S_1 = \{1, 2, 1', 2', 3, 3'\}$ giving the surrogate capacity constraints

$$\begin{array}{llllllllll} v_{21} & +v_{31} & & +v_{41} & & & +5x_1 & & & \geqslant 1 \\ & +v_{31} & +v_{32} & +v_{41} & +v_{42} & & +5x_1 & +5x_2 & & \geqslant 3 \\ & & & v_{41} & +v_{42} & +v_{43} & +5x_1 & +5x_2 & +5x_3 & \geqslant 7 \end{array}$$

leading first to the standard MIR inequalities and then the dynamic inequality

$$v_{21} + v_{31} + v_{32} + v_{41} + v_{42} + v_{43} \geqslant 1(1 - x_1) + (2 - 1)(2 - x_1 - x_2 - x_3) + (3 - 2)(1 - x_1 - x_2).$$

Combinatorial inequalities. With the same structure J, K and K_j of locations and clients, let β be the minimum number of locations required to serve all the clients in K if location j is restricted to serving clients in K_j. Then it is shown in [2] that

$$\sum_{j \in J} \sum_{j \in K_j} \frac{1}{a_i} y_{ij} - \sum_{j \in J} x_j \leqslant |K| - \beta$$

is valid.

4.4. Unit commitment problems

The unit commitment problem (the problem of scheduling electricity generators to satisfy hourly demands for a day or a week) is not a fixed charge network flow problem. However, its formulation as a mixed integer program contains several constraints and variables that have been encountered in this chapter for which cuts can be generated, such as single node flow models and start-up variables linking the generators between time periods. A typical formulation involves the following variables:

x_t^i is the number of generators of type i functioning at period (hour) t (often each generator is distinct, and this is a 0–1 variable)

z_t^i is the increase in the number of generators of type i active in period t

y_t^i is the amount of electricity produced by generators of type i in period t, and as basic constraints

$$\sum_i y_t^i = d_t \quad \text{for all } t, \tag{49}$$

$$l^i x_t^i \leqslant y_t^i \leqslant C^i x_t^i \quad \text{for all } i, t, \tag{50}$$

$$z_t^i \geqslant x_t^i - x_{t-1}^i \quad \text{for all } i, t, \tag{51}$$

$$z_t^i \leqslant x_t^i \quad \text{for all } i, t, \tag{52}$$

$$y_t^i \geqslant 0, \ x_t^i \leqslant u^i \quad \text{for all } i, t, \tag{53}$$

$$x_t^i, \ z_t^i \in Z_+^1 \quad \text{for all } i, t. \tag{54}$$

Typical models also contain ramping and reserve constraints, see [92]. Constraints (49), (50), (53), (54) lead to single node flow sets, or continuous knapsack sets on which various inequalities presented in Section 1 can be generated. In contrast to lot-sizing models, the flow balance constraints are not linked over time, as electricity cannot be stocked. However, the start-up variables provide a certain link between periods. Specifically, if we aggregate (49) for periods $t = k, \dots, l$ and use (50), we obtain $\sum_{t=k}^{l} \sum_i C^i x_t^i \geqslant d_{kl}$. Letting $\chi_{kt}^i = \max\{x_k^i, \dots, x_t^i\}$ and (I_1, I_2) be a partition of the generator set, we obtain

$$\sum_{i \in I_1} C^i \sum_{t=k}^{l} x_t^i + \sum_{i \in I_2} (k - l + 1) C^i \chi_{kl}^i \geqslant d_{kl}$$

with $x_t^i, \chi_{kl}^i \in Z_+^1$. Deriving valid inequalities for such knapsack sets, and then using Observation 4.2 to replace χ_{kl}^i by its upper bound $x_k^i + z_{k+1}^i + \cdots + z_l^i$ leads to new valid inequalities.

Example 4.4 (Marchand [68]). Consider two generator types and two periods with $C^1 = 4, C^2 = 5, d_1 = 12, d_2 = 13$ and $u^1 = u^2 = 4$. Taking $I_1 = \{2\}$ and $I_2 = \{1\}$, we obtain the set

$$5x_1^2 + 5x_2^2 + 8\chi_{12}^1 \geqslant 25, \quad 0 \leqslant x_1^2, x_2^2, \chi_{12}^1 \leqslant 4 \text{ and integer.}$$

A valid inequality for this set is

$$\chi_{12}^1 + x_1^2 + x_2^2 \geqslant 4.$$

Now using $x_1^1 + z_2^1 \geqslant \chi_{12}^1$, we obtain the valid inequality

$$x_1^1 + z_2^1 + x_1^2 + x_2^2 \geqslant 4.$$

This inequality cuts off the extreme point solution $x_1^1 = 3$, $x_2^1 = 3$, $z_2^1 = 0$ and $x_1^2 = 0$, $x_2^2 = \frac{1}{5}$, $z_2^2 = \frac{1}{5}$.

5. Note on computation with cutting planes

Several of the families of valid inequalities described above have been incorporated into branch-and-bound systems in the last fifteen years. If cuts are only added at the top node, we speak of a *cut-and-branch* system, while if cuts are added at other nodes in the enumeration tree, it is a *branch-and-cut* system. Introductions to branch-and-cut can be found in [62,107]. For a survey on branch-and-cut systems for combinatorial optimization problems, see [27,63].

5.1. General mixed integer programming systems

In [36] lifted cover inequalities for 0–1 knapsack inequalities were first incorporated in a cut-and-branch system for 0–1 integer programs. Later flow cover inequalities and an uncapacitated version of the dynamic inequalities on paths were included in MPSARX [101], a cut-and-branch system for MIPs. MINTO [75] was the first branch-and-cut system for MIPs incorporating lifted cover and flow cover inequalities, and more recently lifted cover inequalities for knapsack constraints with generalized upper bound constraints. Computational testing of lifted cover and flow cover inequalities has been reported in [54,53]. More recent systems include SIP [72,73] which also generates feasible set inequalities, and BC-OPT [34] that includes integer knapsack inequalities and recently also MIR inequalities. Taking a different approach, MIPO [11] is a branch-and-cut system for MIPs based on lift-and-project inequalities, where the importance of finding the right balance between cutting and branching is clearly demonstrated. With this system it has also been shown that Gomory mixed integer cuts can be used effectively [12].

Two of the commercial systems, CPLEX and XPRESS, have recently started incorporating lifted cover inequalities, flow cover and MIR inequalities into their systems. For those interested in testing new cuts, etc., a library of mixed integer programming test instances is available [22]. ABACUS [97] is a branch-and-cut framework more specifically suited to combinatorial optimization problems.

5.2. Packing and covering

Most set packing or covering inequalities are used in connection with the solution of set partitioning problems, for instance, [58] exploits clique and cycle inequalities, [23] uses aggregated cycle inequalities in addition. 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 [9], a class of k-projection inequalities by [81], and the mentioned aggregated cycle inequalities by [24], which also apply to set covering. A cutting plane algorithm for set packing problems has been developed in [76]. Note also that clique inequalities are used in many general mixed integer programming systems [35,36,75].

5.3. Network design problems

There is little specialized computational work on single commodity network design problems. However, the cutting planes in the general systems cited above significantly improve performance on some instances. In contrast there has been considerable work on multicommodity problems arising from telecommunications networks. Among others single arc sets [67] and MIR inequalities [26] have been used, and both heuristics [20,21], total enumeration [20] and max cut [15] have been used to generate good cut sets. See also [6,37,38,55].

5.4. Lot-sizing, facility location and other structured MIPs

A variety of multiitem and multilevel lot-sizing problems have been solved using the cutting planes described above, see [32,88,18]. A variety of problem instances are available.[3,4]

[3] Lot-sizing instances available at http://www.eng.auburn.edu/∼gaoyubo.

[4] (http://www.core.ucl.ac.be/wolsey/Lotsizeli.htm), a library of lot-sizing instances.

Some computation on capacitated facility location problems is presented in [1]. The library [17] contains a variety of instances.

Several instances in MIPLIB3.0 are unit commitment instances. For these and other electricity generation applications [74], using knapsack and MIR inequalities significantly improves solution performance [70].

Acknowledgements

We are grateful to K. Aardal for a careful reading of the text.

References

[1] K. Aardal, Capacitated facility location: separation algorithms and computational experience, Math. Programming 81 (1998) 149–175.

[2] K. Aardal, Y. Pochet, L.A. Wolsey, Capacitated facility location: valid inequalities and facets, Math. Oper. Res. 20 (1995) 562–582.

[3] D. Applegate, R.E. Bixby, V. Chvátal, W. Cook, Project-and-lift (a paradigm for finding cuts), Draft, Aussois, March, 1998.

[4] A. Atamturk, On network design cut-set polyhedra, Draft, Department of Industrial Engineering and Operations Research, U.C. at Berkeley, August, 1999.

[5] A. Atamturk, G.L. Nemhauser, M.W.P. Savelsbergh, Conflict graphs in integer programming, Technical Report LEC 98-03, Georgia Institute of Technology, 1998.

[6] A. Balakrishnan, T.L. Magnanti, J. Sokol, Y. Wang, Modeling and solving the single facility line restoration problem, Technical report, MIT, Operations Research Center, 1998. Available at http://web.mit.edu/yiwang/www.

[7] E. Balas, Facets of the knapsack polytope, Math. Programming 8 (1975) 146–164.

[8] E. Balas, Disjunctive programming, Ann. Discrete Math. 5 (1979) 3–51.

[9] E. Balas, A. Ho, Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study, Math. Programming 12 (1980) 37–60.

[10] E. Balas, S. Ceria, G. Cornuéjols, A lift-and-project cutting plane algorithm for mixed 0–1 programs, Math. Programming 58 (1993) 295–324.

[11] E. Balas, S. Ceria, G. Cornuéjols, Mixed 0–1 programming by lift-and-project in a branch-and-cut framework, Management Sci. 42 (1996) 1229–1246.

[12] E. Balas, S. Ceria, G. Cornuéjols, N. Natraj, Gomory cuts revisited, Oper. Res. Lett. 19 (1996) 1–9.

[13] E. Balas, E. Zemel, Lifting and complementing yields all the facets of positive zero-one programming polytopes, Math. Programming (1984) 13–24, Proceedings of the International Congress, Rio de Janeiro, 1981.

[14] E. Balas, E. Zemel, Facets of the knapsack polytope from minimal covers, SIAM J. Appl. Math. 34 (1978) 119–148.

[15] F. Barahona, Network design using cut inequalities, SIAM J Optim. 6 (1996) 823–837.

[16] I. Barany, T.J. Van Roy, L.A. Wolsey, Strong formulations for multi-item capacitated lot-sizing, Management Sci. 30 (1984) 1255–1261.

[17] J.E. Beasley, OR-Library: distributing test problems by electronic mail, J. Oper. Res. Soc. 41 (1990) 1069–1072, available at (http://mscmga.ms.ic.ac.uk/info.html).

[18] G. Belvaux, L.A. Wolsey, BC-PROD: A specialized branch-and-cut system for lot-sizing problems, Management Sci. 46 (2000) 724–738.

[19] C. Berge, Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), in Wissenschaftliche Zeitschrift, Mathematisch-Naturwissenschaftliche Reihe. Martin-Luther-Universität Halle-Wittenberg, 1961.

[20] D. Bienstock, O. Günlük, Capacitated network design—polyhedral structure and computation, ORSA J. Comput. 8 (1996) 243–259.
[21] D. Bienstock, S. Chopra, O. Günlük, C.-Y. Tsai, Minimum cost capacity installation for multicommodity network flows, Math. Programming 81 (1998) 177–199.
[22] R.E. Bixby, S. Ceria, C.M. McZeal, M.W.P. Savelsbergh, An updated mixed integer programming library: MIPLIB 3.0, text and problems available at http://www.caam.rice.edu/∼bixby/miplib/miplib.html
[23] R. Borndörfer, Aspects of set packing, partitioning, and covering, Ph.D. Thesis, Technische Universität Berlin, 1998.
[24] R. Borndörfer, R. Weismantel, Relations among some combinatorial programs, Technical Report Preprint SC 97-54, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1997.
[25] R. Borndörfer, R. Weismantel, Set packing relaxations of some integer programs, Technical Report Preprint SC 97-30, Konrad-Zuse Zentrum für Informationstechnik Berlin, 1997.
[26] B. Brockmüller, O. Günlük, L.A. Wolsey, designing private line networks—polyhedral analysis and computation, Core Discussion Paper 9647, Université Catholique de Louvain, 1996, revised March, 1998.
[27] A. Caprara, M. Fischetti, Branch-and-cut algorithms, in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, Wiley, Chichester, 1997, pp. 45–63.
[28] A. Caprara, M. Fischetti, $\{0, \frac{1}{2}\}$-Chvátal–Gomory cuts, Math. Programming 74 (1996) 221–236.
[29] S. Ceria, C. Cordier, H. Marchand, L.A. Wolsey, Cutting planes for integer programs with general integer variables, Math. Programming 81 (1998) 201–214.
[30] V. Chvátal, Edmonds polytopes and a hierarchy of combinatorial problems, Discrete Math. 4 (1973) 305–337.
[31] V. Chvátal, On certain polytopes associated with graphs, J. Combin. Theory B 18 (1975) 305–337.
[32] M. Constantino, A cutting plane approach to capacitated lot-sizing with start-up costs, Math. Programming 75 (1996) 353–376.
[33] W. Cook, R. Kannan, A. Schrijver, Chvátal closures for mixed integer programming problems, Math. Programming 47 (1990) 155–174.
[34] C. Cordier, H. Marchand, R. Laundy, L.A. Wolsey, bc-opt: A branch-and-cut code for mixed integer programs, bc-opt: a branch-and-cut code for mixed integer programs, Math. Programming 86 (1999) 335–354.
[35] CPLEX, Using the CPLEX callable library, ILOG CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA, Information available at URL http://www.cplex.com (1998).
[36] H. Crowder, E. Johnson, M.W. Padberg, Solving large-scale zero-one linear programming problems, Oper. Res. 31 (1983) 803–834.
[37] G. Dahl, A. Martin, M. Stoer, Routing through virtual paths in layered telecommunication networks, Research Note N78/95, Telenor Research and Development, Kjeller, Norway, 1995, Oper. Res. 49 (1999) 693–702.
[38] G. Dahl, M. Stoer, A cutting plane algorithm for multicommodity survivable network design problems, INFORMS J. Comput. 10 (1998) 1–11.
[39] I.R. de Farias, E.L. Johnson, G.L. Nemhauser, Facets of the complementarity knapsack polytope, Technical Report LEC-98-08, Georgia Institute of Technology, 1998.
[40] J. Edmonds, Matroids and the greedy algorithm, Math. Programming 1 (1971) 127–136.
[41] J. Edmonds, Matroid intersection, Ann. Discrete Math. 4 (1979) 39–49.
[42] R.Euler, M. Jünger, G. Reinelt, Generalizations of odd cycles and anti-cycles and their relation to independence system polyhedra, Math. Oper. Res. 12 (1987) 451–462.
[43] C.E. Ferreira, A. Martin, R. Weismantel, Solving multiple knapsack problems by cutting planes, SIAM J. Optim. 6 (1996) 858–877.
[44] D.R. Fulkerson, Blocking and anti-blocking pairs of polyhedra, Math. Programming 1 (1971) 168–194.
[45] M.X. Goemans, The Steiner tree polytope and related polyhedra, Math. Programming 63 (1994) 157–183.
[46] R.E. Gomory, Outline of an algorithm for integer solutions to linear programs, Bull. Amer. Soc. 64 (1958) 275–278.

[47] R.E. Gomory, An algorithm for the mixed integer problem, Technical Report RM-2597, The RAND Cooperation, 1960.
[48] R.E. Gomory, Solving linear programming problems in integers, in: R. Bellman, M. Hall (Eds.), Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics 10, Providence, RI, 1960.
[49] R.E. Gomory, An algorithm for integer solutions to linear programming, in: R.L. Graves, P. Wolfe (Eds.), Recent Advances in Mathematical Programming, McGraw-Hill, New York, 1969, pp. 269–302.
[50] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer, Berlin, 1988.
[51] M. Grötschel, C.L. Monma, M. Stoer, Design of survivable networks, in: M.O. Ball et al. (Eds.), Network Models, Handbooks in OR and MS 7, Elsevier, Amsterdam, 1995 (chapter 10).
[52] Z. Gu, G.L. Nemhauser, M.W.P. Savelsbergh, Sequence independent lifting in mixed integer programming, J. Combin. Optim. 4 (2000) 109–129.
[53] Z. Gu, G.L. Nemhauser, M.W.P. Savelsbergh, Lifted flow cover inequalities for mixed 0–1 integer programs, Math. Programming A 85 (1999) 436–467.
[54] Z. Gu, G.L. Nemhauser, M.W.P. Savelsbergh, Lifted cover inequalities for 0–1 integer programs: computation, INFORMS J. Comput. 10 (1998) 427–437.
[55] O. Günlük, A branch-and-cut algorithm for capacitated network design, Technical report, Cornell University, 1966.
[56] O. Günlük, Y. Pochet, Mixing mixed-integer inequalities, Math. Programming (2001) 429–458.
[57] P.L. Hammer, E.L. Johnson, U.N. Peled, Facets of regular 0–1 polytopes, Math. Programming 8 (1975) 179–206.
[58] K.L. Hoffman, M.W. Padberg, Solving airline crew-scheduling problems by branch-and-cut, Management Sci. 39 (1993) 657–682.
[59] M. Iri, On an extension of the maximum-flow minimum-cut theorem to multi-commodity flows, J. Oper. Res. Soc. Japan 13 (1970/71) 129–135.
[60] R.G. Jeroslow, Cutting plane theory: disjunctive methods, Ann. Discrete Math. 1 (1977) 293–330.
[61] E.L. Johnson, M.W. Padberg, Degree-two inequalities, clique facets, and biperfect graphs, Ann. Discrete Math. 16 (1982) 169–187.
[62] E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, Progress in linear programming based branch-and-bound algorithms: an exposition, INFORMS J. Comput. 12 (2000) 2–23.
[63] M. Jünger, G. Reinelt, S. Thienel, Practical problem solving with cutting plane algorithms in combinatorial optimization, in: W. Cook, L. Lovász, P. Seymour (Eds.), Combinatorial Optimization, DIMACS Series in Discrete Mathematics and Computer Science, AMS, Providence, RI, 1995, pp. 11–152.
[64] M. Laurent, A generalization of antiwebs to independence systems and their canonical facets, Math. Programming 45 (1989) 97–108.
[65] L. Lovász, On the Shannon capacity of a graph, IEEE Trans. Inform. Theory 25 (1979) 1–7.
[66] L. Lovász, A. Schrijver, Cones of matrices and set-functions and 0–1 optimization, SIAM J. Optim. 1 (1991) 166–190.
[67] T.L. Magnanti, P. Mirchandani, R. Vachani, Modelling and solving the two-facility network loading problem, Oper. Res. 43 (1995) 142–157.
[68] H. Marchand, Etude d'un problème d'optimisation lié à la gestion d'un parc électrique, Engineering Thesis, Faculté des Sciences Appliquées, 1994.
[69] H. Marchand, A polyhedral study of the mixed knapsack set and its use to solve mixed integer programs, Ph.D. Thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1998.
[70] H. Marchand, L.A. Wolsey, Aggregation and mixed integer rounding to solve MIPs, CORE DP9839, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1998, Oper. Res., to appear.
[71] H. Marchand, L.A. Wolsey, The 0–1 knapsack problem with a single continuous variable, Math. Programming 85 (1999) 15–33.
[72] A. Martin, Integer programs with block structure, Habilitations-Schrift Technische Universität Berlin, 1998. Available at (ftp://ftp.zib.de/pub/zib-publications/reports/SC-99-03.ps).
[73] A. Martin, R. Weismantel, The intersection of knapsack polyhedra and extensions, in: R.E. Bixby, E.A. Boyd, R.Z. Rios-Mercado (Eds.), Lecture Notes in Computer Science, Vol. 1412, Springer, Berlin, 1998, pp. 243–256.

[74] MEMIPS, Model enhanced solution methods for integer programming software, Esprit Project 20118, Public Report Reference DR1.1.10, 1997.
[75] G.L. Nemhauser, M.W.P. Savelsbergh, G.C. Sigismondi, MINTO, a mixed INTeger optimizer, Oper. Res. Lett. 15 (1994) 47–58.
[76] G.L. Nemhauser, G. Sigismondi, A strong cutting plane/branch-and-bound algorithm for node packing, J. Oper. Res. Soc. 43 (1992) 443–457.
[77] G.L. Nemhauser, L.E. Trotter, J., Properties of vertex packing and independence system polyhedra, Math. Programming 6 (1974) 48–61.
[78] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiley, New York, 1988.
[79] G.L. Nemhauser, L.A. Wolsey, A recursive procedure to generate all cuts for 0–1 mixed integer programs, Math. Programming 46 (1990) 379–390.
[80] P. Nobili, A. Sassano, Facets and lifting procedures for the set covering polytope, Math. Programming 45 (1989) 111–137.
[81] P. Nobili, A. Sassano, A separation routine for the set covering polytope, in: E. Balas, G. Cornuéjols, R. Kannan (Eds.), Integer Programming and Combinatorial Optimization, Proceedings of the 2nd IPCO Conference, 1992, pp. 201–219.
[82] K. Onaga, O. Kakusho, On feasibility conditions of multicommodity flows in networks, IEEE Trans. Circuit Theory 18 (1971) 425–429.
[83] J.H. Owen, S. Mehrotra, Math. Programming 89 (2001) 437–448.
[84] M.W. Padberg, On the facial structure of set packing polyhedra, Math. Programming 5 (1973) 199–215.
[85] M.W. Padberg, A note on zero-one programming, Oper. Res. 23 (1975) 833–837.
[86] M.W. Padberg, T.J. Van Roy, L.A. Wolsey, Valid linear inequalities for fixed charge problems, Oper. Res. 33 (1985) 842–861.
[87] Y. Pochet, Valid inequalities and separation for capacitated economic lot-sizing, Oper. Res. Lett. 7 (1988) 109–116.
[88] Y. Pochet, L.A. Wolsey, Solving multi-item lot-sizing problems using strong cutting planes, Management Sci. 37 (1991) 53–67.
[89] Y. Pochet, L.A. Wolsey, Lot-sizing with constant batches: formulation and valid inequalities, Math. Oper. Res. 18 (1993) 767–785.
[90] Y. Pochet, L.A. Wolsey, Integer knapsacks and flow covers with divisible coefficients: polyhedra, optimization and separation, Discrete Appl. Math. 59 (1995) 57–74.
[91] R. Rardin, L.A. Wolsey, Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems, European J. Oper. Res. 71 (1993) 95–109.
[92] A. Renaud, Daily generation management at electricité de France: from planning towards real time, IEEE Trans. Automat. Control 38 (1993) 1080–1093.
[93] A. Sassano, On the facial structure of the set covering polytope, Math. Programming 44 (1989) 181–202.
[94] A. Schrijver, On cutting planes, Ann. Discrete Math. 9 (1980) 291–296.
[95] A. Schrijver, Theory of Linear and Integer Programming, Wiley, Chichester, 1986.
[96] H. Sherali, W. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems, SIAM J. Discrete Math. 3 (1990) 411–430.
[97] S. Thienel, ABACUS—a branch-and-cut system, Doctoral Thesis, Universität zu Köln, 1995.
[98] R. van de Leensel, Models and algorithms for telecommunications network design, Proefschrift, Universiteit Maastricht, 1999.
[99] R. van de Leensel, C.P.M. van Hoesel, J.J. van de Klundert, Lifting valid inequalities for the precedence constrained knapsack problem, Math. Programming A 86 (1999) 161–185.
[100] T.J. Van Roy, L.A. Wolsey, Valid inequalities for mixed 0–1 programs, Discrete Appl. Math. 4 (1986) 199–213.
[101] T.J. Van Roy, L.A. Wolsey, Solving mixed 0–1 problems by automatic reformulation, Oper. Res. 35 (1987) 45–57.
[102] R. Weismantel, On the 0/1 knapsack polytope, Math. Programming 77 (1997) 49–68.
[103] L.A. Wolsey, Faces for a linear inequality in 0–1 variables, Math. Programming 8 (1975) 165–178.
[104] L.A. Wolsey, Valid inequalities and superadditivity for 0/1 integer programs, Math. Oper. Res. 2 (1977) 66–77.

[105] L.A. Wolsey, Submodularity and valid inequalities in capacitated fixed charge networks, Oper. Res. Lett. 8 (1989) 119–124.
[106] L.A. Wolsey, Valid inequalities for 0–1 knapsacks and MIPS with generalized upper bound constraints, Discrete Appl. Math. 29 (1990) 251–261.
[107] L.A. Wolsey, Integer Programming, Wiley, New York, 1998.
[108] F. Eisenbrand, On the membership problem for the elementary closure of a polyhedron, Combinatorica 12 (1999) 27–37.

For further reading

A. Balakrishnan, T.L. Magnanti, R.T. Wong, A decomposition algorithm for local access telecommunications network expansion planning, Oper. Res. 43 (1995) 58--76.

W. Cook, L. Lov{á} sz, P. Seymour, Combinatorial Optimization, DI-MACS Series in Discrete Mathematics and Computer Science, AMS, Providence, RI, 1995.

B. Gavish, K. Altinkemer, Backbone network design tools with economic tradeoffs, ORSA J. Comput. 2 (1990) 58--76.

T.I. Magnanti, P. Mirchandani, Shortest paths, single origin-destination network design and associated polyhedra, Networks 23 (1993) 103--121.