Math. Program., Ser. A (2008) 113:241–257
DOI 10.1007/s10107-006-0051-y

FULL LENGTH PAPER

Projected Chvátal–Gomory cuts for mixed integer linear programs

Pierre Bonami · Gérard Cornuéjols · Sanjeeb Dash · Matteo Fischetti · Andrea Lodi

Received: 28 December 2005 / Accepted: 12 October 2006 / Published online: 8 December 2006

Abstract Recent experiments by Fischetti and Lodi show that the first Chvátal closure of a pure integer linear program (ILP) often gives a surprisingly tight approximation of the integer hull. They optimize over the first Chvátal closure by modeling the Chvátal–Gomory (CG) separation problem as a mixed integer linear program (MILP) which is then solved by a general-

Gérard Cornuéjols was supported in part by NSF grant DMI-0352885, ONR grant N00014-03-1-0188, and ANR grant BLAN 06-1-138894.
Matteo Fischetti was supported in part by the EU projects ADONET (contract n. MRTN-CT-2003-504438) and ARRIVAL (contract n. FP6-021235-2).
Andrea Lodi was supported in part by the EU projects ADONET (contract n. MRTN-CT-2003-504438) and ARRIVAL (contract n. FP6-021235-2).

P. Bonami · S. Dash
IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA
e-mail: pbonami@us.ibm.com

S. Dash
e-mail: sanjeebd@us.ibm.com

G. Cornuéjols (✉)
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA

G. Cornuéjols
LIF, Faculté des Sciences de Luminy, 13288 Marseille, France
e-mail: gc0v@andrew.cmu.edu

M. Fischetti
DEI, University of Padova, via Gradenigo 6A, 35131 Padova, Italy
e-mail: matteo.fischetti@unipd.it

A. Lodi
DEIS, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy
e-mail: alodi@deis.unibo.it

purpose MILP solver. Unfortunately, this approach does not extend immediately to the Gomory mixed integer (GMI) closure of an MILP, since the GMI separation problem involves the solution of a nonlinear mixed integer program or a parametric MILP. In this paper we introduce a projected version of the CG cuts, and study their practical effectiveness for MILP problems. The idea is to project first the linear programming relaxation of the MILP at hand onto the space of the integer variables, and then to derive Chvátal–Gomory cuts for the projected polyhedron. Though theoretically dominated by GMI cuts, projected CG cuts have the advantage of producing a separation model very similar to the one introduced by Fischetti and Lodi, which can typically be solved in a reasonable amount of computing time.

Keywords Mixed integer linear program · Chvátal–Gomory cut · Separation problem · Projected polyhedron

Mathematics Subject Classification (2000) 90C10 · 90C11 · 90C57

1 Introduction

Consider first the pure Integer Linear Programming (ILP) problem $\min\{c^T x : Ax \leq b, x \geq 0, x \text{ integral}\}$ where A is an $m \times n$ rational matrix, $b \in \mathbb{Q}^m$, and $c \in \mathbb{Q}^n$, along with the two associated polyhedra $P := \{x \in \mathbb{R}^n_+ : Ax \leq b\}$ and $P_I := conv\{x \in \mathbb{Z}^n_+ : Ax \leq b\} = conv(P \cap \mathbb{Z}^n)$.

A *Chvátal–Gomory* (*CG*) *cut* (also known as *Gomory fractional* cut) [17,8] is an inequality of the form $\lfloor u^T A \rfloor x \leq \lfloor u^T b \rfloor$ where $u \in \mathbb{R}^m_+$ is a vector of multipliers, and $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to its argument. Chvátal–Gomory cuts are valid inequalities for P_I. The *Chvátal closure* of P is defined as

$$P^1 := \{x \geq 0 : \lfloor u^T A \rfloor x \leq \lfloor u^T b \rfloor \text{ for all } u \in \mathbb{R}^m_+\}. \tag{1}$$

Thus $P_I \subseteq P^1 \subseteq P$. By the well-known equivalence between optimization and separation [19], optimizing over the first Chvátal closure is equivalent to solving the *CG separation problem*:

> *CG-SEP*: Given any point $x^* \in P$ find (if any) a CG cut that is violated by x^*, i.e., find $u \in \mathbb{R}^m_+$ such that $\lfloor u^T A \rfloor x^* > \lfloor u^T b \rfloor$, or prove that no such u exists.

It was proved by Eisenbrand [15] that CG-SEP is NP-hard, and therefore so is optimizing over P^1. Fischetti and Lodi [16] recently studied the practical strength of P^1 in approximating P_I. Their approach is to model the CG separation problem as an MILP, which is then solved through a general-purpose MILP solver. To be more specific, given an input point $x^* \in P$ to be separated, CG-SEP calls for a CG cut $\alpha^T x \leq \alpha_0$ which is (maximally) violated by x^*, where $\alpha = \lfloor u^T A \rfloor$ and $\alpha_0 = \lfloor u^T b \rfloor$ for some $u \in \mathbb{R}^m_+$. Hence, if A_j denotes the

jth column of A, CG-SEP can be modeled as:

$$\max \quad \alpha^T x^* - \alpha_0, \tag{2}$$

$$\alpha_j \le u^T A_j, \quad \text{for } j = 1, \ldots, n, \tag{3}$$

$$\alpha_0 + 1 - \epsilon \ge u^T b, \tag{4}$$

$$u_i \ge 0, \quad \text{for } i = 1, \ldots, m, \tag{5}$$

$$\alpha_j \quad \text{integer}, \quad \text{for } j = 0, \ldots, n, \tag{6}$$

where ϵ is a small positive value. The objective function (2) gives the amount of violation of the CG cut evaluated for $x = x^*$, that we want to maximize. Because of the sign of the objective function coefficients, the rounding conditions $\alpha_j = \lfloor u^T A_j \rfloor$ can be imposed through upper bound conditions on variables α_j ($j = 1, \ldots, n$), as in (3), and with a lower bound condition on α_0, as in (4). Note that this latter constraint requires the introduction of a small value ϵ to ensure that when $u^T b$ integral, $\alpha_0 = u^T b$ and not $u^T b - 1$.

Model (2)–(6) can also be explained by observing that $\alpha^T x \le \alpha_0$ is a CG cut if and only if (α, α_0) is an integral vector, as stated in (6), and $\alpha^T x \le \alpha_0 + 1 - \epsilon$ is a valid inequality for P, as stated in (3)–(5) by using the well-known characterization of valid inequalities for a polyhedron due to Farkas.

Unfortunately, model (2)–(6) does not extend immediately to the mixed integer case, where one typically concentrates on the stronger Gomory Mixed Integer (GMI) cuts [18]. Although it is easy to find a GMI cut that separates an integer infeasible *basic solution* of the linear programming relaxation, separating other points by GMI cuts is NP-hard [7,13]. Define the *Gomory mixed integer closure* as the intersection of all the GMI cuts with the nonnegative orthant. Not only is the separation problem for the Gomory mixed integer closure NP-hard, but there is no MILP model like (2)–(6) known for it. Indeed, one faces the solution of a nonlinear [7,14] or parametric [4] mixed integer problem for the separation of GMI cuts. In this paper we introduce a projected version of the classical CG cuts, and study their strength on general instances of MILP as well as on some specific classes. The idea is to project first the linear programming relaxation of the MILP at hand onto the space of the integer variables, and then to derive Chvátal–Gomory cuts for the projected polyhedron. Though theoretically dominated by GMI cuts, projected CG cuts have the advantage of producing an MILP separation model very similar to (2)–(6), hence its solution can typically be carried out in a reasonable amount of computing time. Also, it can be conjectured that projected CG cuts are more "combinatorial" in nature than GMI cuts, and can be quite effective for a large class of MILPs—notably, those where the continuous variables are only used to model some feasibility condition, possibly by using big-M coefficients, and are not present in the objective function, as, e.g., those addressed in [9].

The present paper is organized as follows. In Sect. 2 we define more precisely our projected CG cuts, and give an MILP formulation of the associated separation problem. In Sect. 3, we prove a theorem showing that projected CG cuts are equivalent to split cuts [11] in which one term of the disjunction has an empty intersection with the original formulation. In Sect. 4 we consider classes of problems where projected CG cuts are likely to be effective. Computational results on all the *mixed* MILP instances taken from the MIPLIB 3.0 library [5] are presented in Sect. 5, as well as on instances of the asymmetric traveling salesman with time windows. These results show the effectiveness of projected CG cuts both on general instances and on instances arising in specific contexts. Section 6 contains concluding remarks and future directions of research.

2 Projected Chvátal–Gomory cuts

The computational results reported in [16] show that P^1 often gives a surprisingly tight approximation of P_I, so a natural question is whether the same result generalizes to mixed integer linear programming problems.

In this paper, we consider a Mixed Integer Linear Program (MILP) of the form

$$\min\{c^T x + f^T y : Ax + Cy \leq b, x \geq 0, x \text{ integral}, \; y \geq 0\}, \tag{7}$$

where A and C are $m \times n$ and $m \times r$ rational matrices respectively, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$, and $f \in \mathbb{Q}^r$. We also consider the two following polyhedra in the (x, y)-space:

$$P(x, y) := \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^r_+ : Ax + Cy \leq b\}, \tag{8}$$

$$P_I(x, y) := conv(\{(x, y) \in P(x, y) : x \text{ integral}\}). \tag{9}$$

Our first order of business is to extend the classical definition of Chvátal–Gomory cuts to the mixed integer case, in such a way that the resulting separation problem retains as much as possible the simple structure of model (2)–(6). To this end, we define the projection of $P(x, y)$ onto the space of the x variables as:

$$\begin{aligned} P(x) \; &:= \{x \in \mathbb{R}^n_+ : \text{there exists } y \in \mathbb{R}^r_+ \text{ s.t. } Ax + Cy \leq b\} && (10)\\ &= \{x \in \mathbb{R}^n_+ : u^k A x \leq u^k b, \; k = 1, \ldots, K\} && (11)\\ &=: \{x \in \mathbb{R}^n_+ : \bar{A}x \leq \bar{b}\}, && (12) \end{aligned}$$

where $u^1, \ldots, u^K$ are the (finitely many) extreme rays of the projection cone $\{u \in \mathbb{R}^m_+ : u^T C \geq 0^T\}$. Note that the rows of the linear system $\bar{A}x \leq \bar{b}$ are of Chvátal rank 0 with respect to $P(x, y)$, i.e, no rounding argument is needed to prove their validity.

We then define a *projected Chvátal–Gomory (pro-CG) cut* as a CG cut derived from the system $\bar{A}x \leq \bar{b}$, $x \geq 0$, i.e., an inequality of the form $\lfloor w^T \bar{A} \rfloor x \leq \lfloor w^T \bar{b} \rfloor$ for some $w \geq 0$. We denote by $P^1(x)$ the Chvátal closure of $P(x)$ and by $P_I(x)$ the convex hull of $P(x) \cap \mathbb{Z}^n$. Since any row of $\bar{A}x \leq \bar{b}$ can be obtained as a linear combination of the rows of $Ax \leq b$ with multipliers

$\bar{u} \geq 0$ such that $\bar{u}^T C \geq 0^T$, it follows that a pro-CG cut can equivalently (and more directly) be defined as an inequality of the form

$$\lfloor u^T A \rfloor x \leq \lfloor u^T b \rfloor \quad \text{for any } u \geq 0 \text{ such that } u^T C \geq 0^T. \tag{13}$$

As such, its associated separation problem can be modeled as a simple extension of (2)–(6), through the following MILP:

$$\max \ \alpha^T x^* - \alpha_0, \tag{14}$$

$$\alpha_j \leq u^T A_j, \quad \text{for } j = 1, \ldots, n, \tag{15}$$

$$0 \leq u^T C_j, \quad \text{for } j = 1, \ldots, r, \tag{16}$$

$$\alpha_0 + 1 - \epsilon \geq u^T b, \tag{17}$$

$$u_i \geq 0, \quad \text{for } i = 1, \ldots, m, \tag{18}$$

$$\alpha_j \ \text{integer}, \quad \text{for } j = 0, \ldots, n. \tag{19}$$

3 Connection with split cuts

In this section, we relate the projected Chvátal–Gomory cuts to known cuts for MILP. For this, it will be convenient to define the *Chvátal–Gomory closure* of $P(x, y)$ as the intersection of $P(x, y)$ with all the pro-CG cuts (viewed as inequalities $\alpha^T x + 0^T y \leq \alpha_0$ in $\mathbb{R}^n \times \mathbb{R}^r$). We denote the Chvátal–Gomory closure of $P(x, y)$ by $P^1(x, y)$. Since the intersection of all pro-CG cuts is a polyhedron, it follows that so is $P^1(x, y)$.

Split cuts were introduced by Cook, Kannan and Schrijver [11]. They are obtained as follows. For any $\pi \in \mathbb{Z}^n$ and $\pi_0 \in \mathbb{Z}$, the disjunction $\pi^T x \leq \pi_0$ or $\pi^T x \geq \pi_0 + 1$ is valid for MILP. In other words, $P_I(x, y) \subseteq conv(\Pi_0 \cup \Pi_1)$ where

$$\Pi_0 := P(x, y) \cap \{(x, y) : \pi^T x \leq \pi_0\}, \tag{20}$$
$$\Pi_1 := P(x, y) \cap \{(x, y) : \pi^T x \geq \pi_0 + 1\}. \tag{21}$$

A valid inequality for $conv(\Pi_0 \cup \Pi_1)$ is called a *split cut*. The *split closure* of $P(x, y)$ is the convex set obtained by intersecting all the split cuts for all $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$. Cook, Kannan and Schrijver proved that the split closure of $P(x, y)$ is a polyhedron. Nemhauser and Wolsey [24] proved that the split closure and the Gomory mixed integer closure are identical sets. See [12] for a direct proof of this result. Projected Chvátal–Gomory cuts are dominated by GMI cuts, and therefore $P^1(x, y)$ contains the split closure of $P(x, y)$. The following result gives the precise relation between the two classes of cuts.

Theorem 1 *Let $S(x, y)$ denote the intersection of $P(x, y)$ with all the split cuts for all $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$ where one of the sets Π_0, Π_1 defined in* (20) *and* (21) *is empty.*

Then

$$P^1(x,y) = S(x,y).$$

Proof First we prove $S(x,y) \subseteq P^1(x,y)$. Consider an inequality that defines a facet of $P^1(x,y)$. If it is valid for $P(x,y)$, then it is clearly valid for $S(x,y)$. So we may assume that the facet of $P^1(x,y)$ is defined by a pro-CG cut $\pi^T x \leq \pi_0$. By the Chvátal–Gomory procedure $\pi^T x \leq \beta$ must be a valid inequality for $P(x,y)$ for some $\beta < \pi_0 + 1$. This implies that $\Pi_1 := P(x,y) \cap \{(x,y) : \pi^T x \geq \pi_0 + 1\}$ is empty. Therefore $conv(\Pi_0 \cup \Pi_1) = \Pi_0$. This implies that $\pi^T x \leq \pi_0$ is valid for $conv(\Pi_0 \cup \Pi_1)$, proving that it is a split cut. Furthermore this split cut is valid for $S(x,y)$ since $\Pi_1 = \emptyset$.

Conversely, we prove $P^1(x,y) \subseteq S(x,y)$. Consider a valid inequality for $S(x,y)$. If it is valid for $P(x,y)$, then it is clearly valid for $P^1(x,y)$. So we only need to consider a valid inequality for $S(x,y)$ that arises from a split cut where one of the sets Π_0, Π_1 is empty, for some $\pi \in \mathbb{Z}^n$ and $\pi_0 \in \mathbb{Z}$. Without loss of generality we may assume that $\Pi_1 = \emptyset$. In other words, the inequality under consideration is valid for Π_0. We will show that $P^1(x,y) \subseteq \Pi_0$. Since all the inequalities that define Π_0 are valid for $P(x,y)$ except possibly for the inequality $\pi^T x \leq \pi_0$, it suffices to show that $\pi^T x \leq \pi_0$ is a pro-CG cut. Let

$$\begin{aligned} \beta = \max \ & \pi^T x \\ & x \in P(x,y). \end{aligned}$$

Since $\Pi_1 = \emptyset$, it follows that $\beta < \pi_0 + 1$. Therefore $\pi^T x \leq \beta$ is a valid inequality for $P(x,y)$. Since y does not appear in this inequality, it is also valid for $P(x)$. The Chvátal–Gomory procedure implies that $\pi^T x \leq \lfloor \beta \rfloor \leq \pi_0$ is a pro-CG cut.

4 On the strength of projected CG cuts

In this section we address the issue of the expected strength of the projected CG cuts. For this it is useful to distinguish between two extreme cases of MILPs: those where the essence of the problem is in the optimization of the integer variables x, and those where optimizing over the continuous variables is the key. This can be illustrated by the following simple example in two variables x and y (with x integer and y continuous): $P(x,y)$ is the polytope defined by the inequalities $x + y \leq 3/2$, $y \leq x$ and $x, y \geq 0$. Observe that the pro-CG cut $x \leq 1$ cuts off the integer infeasible vertex $(3/2, 0)$, but there is no pro-CG cut which cuts off the integer infeasible vertex $(3/4, 3/4)$. Thus, if the objective is to maximize x, pro-CG cuts help, and optimizing over $P^1(x) = P_I(x)$ yields the optimal solution. On the other hand, if the objective is to maximize y, pro-CG cuts do not help in this example. More generally, suppose that the projection x^* of the optimum (x^*, y^*) of the MILP relaxation belongs to the first Chvátal closure $P^1(x)$. In this case, no pro-CG cut can cut off that point, although there might possibly be a huge gap between the MILP and its relaxation.

On the other hand, pro-CG cuts are well suited to handle those MILPs where the continuous variables are only used to model some feasibility condition, possibly by using big-M coefficients, but are not present in the objective function. Indeed, take any inequality of the form $g^T x + 0^T y \leq g_0$ that is valid for $P_I(x, y)$. Then $g^T x \leq g_0$ is also a valid inequality for the projected integer polyhedron $P_I(x)$, hence it is of finite Chvátal rank, say q, with respect to system $\bar{A}x \leq \bar{b}, x \geq 0$ (Chvátal [8], Gomory [18]). This implies that $g^T x \leq g_0$ is a pro-CG cut (of the same rank q) with respect to the original system $Ax + Cy \leq b, (x, y) \geq 0$. In particular, if $f = 0$ in the objective function of MILP (7) and z^* denotes the optimum objective value of MILP, the inequality $c^T x + f^T y \geq z^*$ is valid for $P_I(x, y)$ and therefore it is a pro-CG cut of finite rank. We have just proved the following result.

Theorem 2 *MILPs where the continuous variables do not appear in the objective function can be optimized to proven optimality by using only pro-CG cuts* (*in an iterative way of course*).

A class of problems where (even rank 1) pro-CG cuts are likely to be really effective has been recently addressed by Codato and Fischetti [9]. These authors considered a basic 0-1 ILP of the form

$$\min\{c^T x : Fx \leq g, \quad x \in \{0,1\}^n\} \tag{22}$$

amended by a set of "conditional" linear constraints involving additional continuous variables y, of the form

$$x_i = 1 \Rightarrow w_i^T y \leq w_{i0}, \quad \text{for all } i \in I \tag{23}$$

plus a (possibly empty) set of k (say) "unconditional" constraints on the continuous variables y, namely

$$Dy \leq d. \tag{24}$$

Note that the continuous variables y do not appear in the objective function—they are only introduced to force some feasibility properties of the x's. A familiar example of a problem of this type is the classical Asymmetric Traveling Salesman Problem (ATSP) with time windows, called TW-ATSP in the sequel. Here the binary variables x_{ij} are the usual arc variables, and the continuous variables y_i give the arrival time at city i. Each arc (i, j) has duration d_{ij}, and each city i has to be visited within the time window $[e_i, l_i]$. For this problem, the basic formulation (22) contains the standard ATSP out- and in-degree equations (plus any other ATSP constraints such as subtour elimination etc.). Implications (23) are of the form

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + d_{ij} \tag{25}$$

whereas (24) bounds the arrival time at city i

$$e_i \leq y_i \leq l_i \text{ for all } i \in I. \tag{26}$$

Another example is the map labeling problem [21], where the binary variables are associated with the relative position of two labels to be placed on a map, the continuous variables give their placement coordinates, and the conditional constraints impose non-overlapping conditions of the type "if label i is placed on the right of label j, then the placement coordinates of i and j must obey a certain linear inequality giving a suitable separation condition".

The usual way implications (23) are modeled within the MILP framework is to use the famous *big-M method*, where large positive coefficients M are introduced to activate/deactivate the conditional constraints to be added to the basic model (22), as in:

$$w_i^T y + M(x_i - 1) \leq w_{i0} \text{ for all } i \in I. \tag{27}$$

For example, the TW-ATSP implications (25) are usually modeled as:

$$y_i - y_j + Mx_{ij} \leq M - d_{ij}. \tag{28}$$

It is known however that, due to the presence of the big-M coefficients, the LP relaxation of the resulting MILP model is typically very poor. As a matter of fact, the x solutions of the LP relaxation are only marginally affected by the addition of the y variables and of the associated constraints. To remedy this behavior, Codato and Fischetti proposed the use of the so-called *Combinatorial Benders'* (CB) cuts:

$$\sum_{i \in Q} x_i \leq |Q| - 1, \tag{29}$$

where $Q \subseteq I$ induces a minimal (irreducible) infeasible subsystem of (23)–(24), i.e., an inclusion-minimal set of row-indices of system (23) such that the linear subsystem

$$w_i^T y \leq w_{i0}, \quad \text{for all } i \in Q, \tag{30}$$

$$Dy \leq d \tag{31}$$

has no feasible (continuous) solution y. In a sense, CB cuts try to project in a purely combinatorial way the feasibility requirement in the x space (hence their name). They can be viewed as an attempt to distill automatically some combinatorial information from the input MILP model. In this process, the role of the big-M terms in the MILP model vanishes—only implications (23) are relevant, no matter how they are modeled. The computational results reported in [9] show that CB cuts can be really effective for specific classes of MILPs that are notoriously very hard to solve: even with a simple implementation of the CB cut separation procedure, the use of CB cuts results in a speed-up by several orders of magnitude compared to the best commercial MILP solvers on some important classes of MILPs.

The next proposition shows that CB cuts are a special case of projected CG cuts.

Theorem 3 *Combinatorial Benders cuts are projected CG cuts.*

Proof Consider a combinatorial Benders cut $\sum_{i \in Q} x_i \leq |Q| - 1$ where Q induces a minimal infeasible system of (23)–(24). Maximizing $\sum_{i \in Q} x_i$ over the feasible region $P(x, y)$ of the big-M MILP yields an objective value $\beta < |Q|$, since all x_i cannot be 1. Therefore the Chvátal–Gomory procedure implies that $\sum_{i \in Q} x_i \leq |Q| - 1$ is a CG cut for $P(x, y)$. Since the y variables do not appear in $\sum_{i \in Q} x_i \leq |Q| - 1$, it is also a projected CG cut.

Projected CG cuts can however be much stronger than CB cuts, in that they can exploit all the information contained in the basic model (22). We illustrate this through the TW-ATSP example. Suppose you have a simple dipath P of cardinality k (say) from a certain node a to a certain node b, whose total duration exceeds the difference $l_b - e_a$. To fix the ideas, let the dipath be $P := \{(0, 1), (1, 2), (2, 3), (3, 4)\}$, hence $k = 4$, and let $d_{ij} = 10$ for all $(i, j) \in P$, with $e_0 = 5$ and $l_4 = 40$. The TW-ATSP model includes the following constraints (we choose $M = 100$), plus the nonnegativity constraints on the x variables:

out0: $x_{01} + x_{02} + x_{03} + x_{04} \leq 1$
out1: $x_{10} + x_{12} + x_{13} + x_{14} \leq 1$
out2: $x_{20} + x_{21} + x_{23} + x_{24} \leq 1$
out3: $x_{30} + x_{31} + x_{32} + x_{34} \leq 1$

in1: $x_{01} + x_{21} + x_{31} + x_{41} \leq 1$
in2: $x_{02} + x_{12} + x_{32} + x_{42} \leq 1$
in3: $x_{03} + x_{13} + x_{23} + x_{14} \leq 1$
in4: $x_{04} + x_{14} + x_{24} + x_{34} \leq 1$

t01: $y_0 - y_1 + 100x_{01} \leq 90$
t12: $y_1 - y_2 + 100x_{12} \leq 90$
t23: $y_2 - y_3 + 100x_{23} \leq 90$
t34: $y_3 - y_4 + 100x_{34} \leq 90$
early0: $-y_0 \leq -5$
late4: $y_4 \leq 40$

Clearly, every feasible TW-ATSP solution has to satisfy the *infeasible path constraint*

$$x(P) := \sum_{(i,j) \in P} x_{ij} \leq |P| - 1, \text{ i.e., } x_{01} + x_{12} + x_{23} + x_{34} \leq 3$$

in our case. This cut is a CB cut, since clearly P induces an infeasible subset of system (25)–(26). Because of the discussion above, the cut is also a projected CG cut. This can easily be verified by maximizing the left-hand-side of the cut (namely, $x_{01} + x_{12} + x_{23} + x_{34}$) over the above system of linear constraints, obtaining an optimal value of 3.95 (to be rounded down to 3). However, the path infeasibility constraint is rather weak in that it does not take into account the presence of the out- and in-degree constraints, as in the stronger *tournament inequality* $x([P]) \leq |P| - 1$ proposed by Ascheuer, Fischetti and Grötschel [2], where P is any infeasible path, and $[P] := \{(i,j) : \text{node } i \text{ precedes node } j \text{ in } P\}$ is its transitive closure. In our example, the tournament inequality reads

$$x_{01} + x_{02} + x_{03} + x_{04} + x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} \leq 3.$$

Optimizing the left-and-side over the LP system above produces an optimal solution value of 3.9875 (still rounded down to 3) showing that the tournament inequality is a projected CG cut.

5 Computational results

In this section we report the outcome of our experiments on a test-bed made up of 43 mixed-integer problems from MIPLIB 3.0 [5]. The approach follows the scheme used in [16], i.e., we implemented a pure cutting plane algorithm where, at each iteration, pro-CG cuts are generated by solving the separation problem (14)–(19) through a standard MILP solver. In order to speedup the overall computation, the MILP solver is aborted whenever its incumbent solution does not improve for a certain number of branching nodes (100 nodes when the cut violation is greater than 0.2 and 1000 nodes otherwise). Our implementation of the cutting-plane method uses the commercial software `ILOG-Cplex` 9.0 as the LP solver, whereas the separation problem is solved through `ILOG-Cplex` 9.0 MILP solver with "mip emphasis 4" parameter; see [20]. All computing times refer to a 3.2 Ghz Pentium 4 PC with 2 GB of RAM.

In particular, Table 1 reports the results for the cutting plane algorithm using pro-CG cuts while Tables 2–3 compare those results with other general-purpose cuts.

Table 1 is partitioned into three parts: at the top we report 10 instances for which we have been able to optimize over the Chvátal–Gomory closure in the time limit of 20 CPU minutes (1,200 CPU seconds[1]), then we have 26 instances in which our cutting plane procedure exceeded such a time limit, and finally, we report 7 instances for which the algorithm did not find any pro-CG cut and proved that none exists. More precisely, on the 10 instances in which our algorithm completes the optimization over the pro-CG closure, no time/node limit is imposed on the solution of the separation MIP which is however optimized—without finding any feasible solution, i.e., any violated pro-CG inequality—with some degree of tolerance. For each instance, we report besides its name (instance), the numbers of integer (n) and continuous variables (r) and the number of continuous variables with a nonzero coefficient in the objective function (r_c). Then, we report for the pro-CG cuts, the number of iterations and the number of separated cuts, the CPU time and the percentage of gap closed computed as $100\frac{\text{opt\_value}(P^1)-\text{opt\_value}(P)}{\text{opt\_value}(P_I)-\text{opt\_value}(P)}$. For those instances for which

[1] Such a time limit has been selected to give a flavor of the practical usefulness of pro-CG cuts, where this is an amount of time a user might reasonably give.

Table 1 Mixed integer linear programs of the MIPLIB 3.0. Note that for instances `dsbmip` and `noswot` there is no gap between the initial LP (though fractional) solution and the optimal value, while for the optimal solution of instance `arki001` we used the best known of value 7,580,877.1907

Instance	n	r	r_c	Pro-CG			
				No. of iterations	No. of cuts	CPU time	Percentage of gap closed
bell3a	71	62	46	70	241	65.3	48.10
bell5	58	46	32	36	126	4.4	91.73
egout	55	86	55	35	168	6.8	81.77
fixnet6	378	500	416	34	83	42.9	67.51
khb05250	24	1,326	1,249	5	13	3.5	4.70
noswot	100	28	0	39	118	68.0	–
rentacar	55	9,502	177	7	15	5.1	0.00
set1ch	240	472	232	29	89	34.2	51.41
vpm1	168	210	0	27	53	14.9	100.00
vpm2	168	210	0	89	275	1,021.9	62.86
10teams	1,800	225	225	455	2,001	1,200.0	$\geq$ 57.14
arki001	538	850	1	62	215	1,200.0	$\geq$ 28.04
blend2	264	89	0	363	1,032	1,200.0	$\geq$ 36.40
dano3mip	552	13,321	1	1	0	1,200.0	$\geq$ 0.00
danoint	56	465	1	4	3	1,200.0	$\geq$ 0.01
dcmulti	75	473	473	46	132	1,200.0	$\geq$ 47.25
dsbmip	192	1,694	1,068	186	433	1,200.0	–
fiber	1,254	44	0	289	1,556	1,200.0	$\geq$ 4.83
flugpl	11	7	7	3	2	1,200.0	$\geq$ 19.19
gen	150	720	432	171	427	1,200.0	$\geq$ 86.60
gesa2	408	816	624	383	1,660	1,200.0	$\geq$ 94.84
gesa2_o	720	504	312	76	306	1,200.0	$\geq$ 94.93
gesa3	384	768	528	138	381	1,200.0	$\geq$ 58.96
gesa3_o	672	480	264	49	193	1,200.0	$\geq$ 64.53
markshare1	50	12	12	3,345	20,686	1,200.0	$\geq$ 0.00
markshare2	60	14	14	3,111	18,720	1,200.0	$\geq$ 0.00
mkc	5,323	2	0	87	267	1,200.0	$\geq$ 1.27
misc03	159	1	1	303	852	1,200.0	$\geq$ 34.92
misc07	259	1	1	331	889	1,200.0	$\geq$ 3.86
pp08a	64	176	112	7	8	1,200.0	$\geq$ 4.32
pp08aCUTS	64	176	112	4	5	1,200.0	$\geq$ 0.68
qiu	48	792	264	7	8	1,200.0	$\geq$ 10.71
qnet1	1,417	124	124	214	715	1,200.0	$\geq$ 7.32
qnet1_o	1,417	124	124	318	1,340	1,200.0	$\geq$ 8.61
rout	315	241	1	459	1,715	1,200.0	$\geq$ 0.03
swath	6,724	81	1	354	1,222	1,200.0	$\geq$ 7.68
mas74	150	1	1	1	0	0.0	0.00
mas76	150	1	1	1	0	0.0	0.00
misc06	112	1,696	1	1	0	0.0	0.00
mod011	96	10,862	7,489	1	0	0.4	0.00
modglob	98	324	324	1	0	0.0	0.00
pk1	55	31	1	1	0	0.0	0.00
rgn	100	80	80	1	0	0.6	0.00

the imposed time limit has been reached the gap closed is reported with a "≥" symbol to indicate that such a gap is indeed underestimated.

The results in Table 1 show that the projected Chvátal–Gomory closure can be an effective approximation of the integer hull of MILPs. The average gap closed over 41 instances[2] is around 29%. On the other hand, as expected, there are several (at least 7) instances for which no pro-CG cut exists. For 11 instances out of 41, optimizing over the projected Chvátal–Gomory closure (up to the time limit of 1,200 s) produced absolutely no improvement. For the 30 remaining instances, however, the average gap closed is around 40%. On certain instances (`bell5, gesa2`), the projected Chvátal–Gomory closure closes over 90% of the gap. On `vpm1` the projected Chvátal–Gomory closure even closes 100% of the gap. This is impressive considering that the pro-CG cuts are also attractive from a numerical point of view: when used iteratively, they tend to deteriorate less rapidly than the GMI cuts read from the LP tableau.

In Tables 2 and 3 we report comparisons with classical families of cutting planes that are valid for the Gomory mixed integer closure: Gomory Mixed Integer cuts from the optimal tableau of the LP relaxation, MIR cuts (Marchand and Wolsey [22]) and lift-and-project cuts [3]. Specifically, the columns GMI and MIR in Table 2 refer to one round of Gomory Mixed Integer cuts, and of Mixed Integer Rounding cuts respectively, as implemented in the COIN-OR cut generator [10]. The column L&P in Table 3 refers to the gap closed by the lift-and-project closure plus a strengthening step, as implemented by Bonami and Minoux [6]. Note that we set a time limit of 20 CPU minutes on each run: 5 instances were interrupted because of the time limit. Tables 2 and 3 show the improvement achieved by the projected Chvátal–Gomory closure when it is applied subsequently to the three other families of cuts, either separately, or all together (the column GMI+MIR+L&P was obtained by applying first the GMI and MIR cuts and then, starting from the resulting solution, the L&P separation step). An additional time limit of 20 CPU minutes was set on generating projected Chvátal–Gomory cuts, for all the runs. Note that for instances where we only partially optimize over the projected Chvátal–Gomory closure, it can happen that the pro-CG gap closed is better than the GMI+pro-CG gap closed (`blend2` is such an instance). We can make the following observations. The pro-CG cuts can sometimes be vastly superior to the other families of cuts (`bell5, gesa2, vpm1`). The average gap closed by the projected Chvátal–Gomory closure (29%) is comparable to that closed by GMI cuts (24%), MIR cuts (23%) and the lift-and-project closure (35%). Tables 2 and 3 show that pro-CG cuts are quite different from the other families of cuts. Adding the pro-CG cuts to the GMI cuts improves the average closed gap from 24% to 41%. Adding them to MIR cuts improves it from 23% to 40%, and adding them to the lift-and-project closure improves it from 35% to 49%. Finally, adding the pro-CG cuts to all the other cuts combined still improves the average gap from 48% to 55%. The case of `egout` is interesting: the gap is closed completely by

[2] Instances `dsbmip` and `noswot` are not considered in the average.

Table 2 Comparison with GMI cuts and MIR cuts

Instance	Percentage of gap closed			
	GMI	GMI + Pro-CG	MIR	MIR + Pro-CG
bell3a	45.10	78.71	19.06	60.94
bell5	14.53	92.64	0.40	91.88
egout	40.26	84.18	57.14	92.74
fixnet6	10.27	75.96	69.92	82.48
khb05250	74.91	74.91	77.92	77.92
noswot	–	–	–	–
rentacar	0.00	0.00	0.00	0.00
set1ch	38.11	70.41	38.27	69.41
vpm1	10.00	100.00	33.08	100.00
vpm2	13.00	64.70	31.52	68.55
10teams	100.00	100.00	0.00	≥57.14
arki001	34.72	≥36.01	7.03	≥33.19
blend2	16.29	≥31.39	14.39	≥32.06
dano3mip	0.01	≥0.01	0.01	≥0.01
danoint	0.22	≥0.22	0.49	≥0.49
dcmulti	47.25	≥67.88	7.49	≥54.23
dsbmip	–	–	–	–
fiber	72.18	≥75.53	25.27	≥30.12
flugpl	11.74	≥11.74	0.00	≥19.19
gen	55.11	≥91.52	57.17	≥93.13
gesa2	30.89	≥98.04	60.69	≥96.49
gesa2_o	31.02	≥98.09	24.62	≥96.54
gesa3	45.76	≥62.99	65.28	≥72.24
gesa3_o	49.16	≥69.98	60.06	≥71.03
markshare1	0.00	≥0.00	0.00	≥0.00
markshare2	0.00	≥0.00	0.00	≥0.00
mkc	13.82	≥14.11	0.00	≥0.01
misc03	8.62	≥30.32	0.00	≥35.11
misc07	0.72	≥4.12	0.00	≥4.12
pp08a	52.10	≥52.31	60.16	≥60.44
pp08aCUTS	29.73	≥30.48	79.55	≥79.59
qiu	0.27	≥7.85	0.00	≥10.71
qnet1	10.57	≥14.41	21.06	≥25.50
qnet1_o	44.49	≥47.12	48.33	≥51.17
rout	0.32	≥0.32	0.00	≥0.12
swath	3.06	≥10.53	0.00	≥7.92
mas74	6.67	6.67	4.14	4.14
mas76	6.42	6.42	5.15	5.15
misc06	30.39	30.39	0.00	0.00
mod011	1.67	1.67	0.10	0.10
modglob	16.85	16.85	13.22	13.22
pk1	0.00	0.00	0.00	0.00
rgn	1.61	1.61	34.21	34.21

combining the 4 types of cuts but not without the pro-CG cuts. Other interesting cases are `bell3a` and `flugpl`, where the pro-CG cuts improve greatly over all the other cuts combined. This indicates that the pro-CG cuts are genuinely

Table 3 Comparison with lift-and-project cuts and a combination of cuts. Note that for vpm2 and misc06, the gap closed by L&P is larger than for GMI+MIR+L&P. This happens because the L&P cuts are strengthened and therefore there is no domination property

Instance	Percentage of gap closed			
	L&P	L&P +pro-CG	GMI +MIR +L&P	GMI +MIR +L&P +pro-CG
bell3a	43.76	81.47	64.02	91.68
bell5	83.25	92.82	85.40	93.18
egout	93.83	98.84	93.85	100.00
fixnet6	85.38	91.96	86.01	92.33
khb05250	99.39	99.39	98.43	98.43
noswot	–	–	–	–
rentacar	≥ 0.00	0.00	≥ 0.00	0.00
set1ch	39.96	68.88	40.17	69.27
vpm1	31.40	100.00	53.90	100.00
vpm2	54.28	79.05	35.48	69.22
10teams	0.00	≥ 57.14	100.00	100.00
arki001	34.13	≥ 34.13	66.67	≥ 79.19
blend2	21.56	≥ 35.86	21.71	≥ 33.44
dano3mip	≥ 0.00	≥ 0.00	≥ 0.01	≥ 0.01
danoint	≥ 1.57	≥ 1.57	≥ 1.61	≥ 1.61
dcmulti	97.22	≥ 97.30	97.65	≥ 97.95
dsbmip	–	–	–	–
fiber	81.68	≥ 83.30	89.68	≥ 91.39
flugpl	0.00	≥ 19.19	11.74	≥ 41.75
gen	78.65	≥ 92.29	81.54	≥ 97.05
gesa2	37.83	≥ 96.69	81.55	≥ 99.21
gesa2_o	37.83	≥ 98.60	49.27	≥ 99.27
gesa3	11.21	≥ 58.30	68.04	≥ 71.05
gesa3_o	11.21	≥ 63.19	68.12	≥ 74.79
markshare1	0.00	≥ 0.00	0.00	≥ 0.00
markshare2	0.00	≥ 0.00	0.00	≥ 0.00
mkc	≥ 26.82	≥ 29.07	≥ 36.65	≥ 39.35
misc03	39.67	≥ 44.91	40.21	≥ 42.70
misc07	12.03	≥ 12.03	12.25	≥ 12.25
pp08a	80.46	≥ 80.46	81.35	≥ 81.35
pp08aCUTS	69.36	≥ 69.36	88.87	≥ 88.87
qiu	0.00	≥ 10.85	28.79	≥ 28.95
qnet1	6.61	≥ 11.99	28.26	≥ 31.54
qnet1_o	0.00	≥ 8.61	48.39	≥ 50.77
rout	30.09	≥ 31.17	30.51	≥ 31.57
swath	≥ 0.32	≥ 8.41	≥ 17.79	≥ 21.50
mas74	0.00	0.00	6.84	6.84
mas76	0.00	0.00	7.03	7.03
misc06	79.52	79.52	46.51	46.51
mod011	5.08	5.08	16.23	16.23
modglob	57.08	57.08	60.76	60.76
pk1	0.00	0.00	0.00	0.00
rgn	79.49	79.49	96.14	96.14

Table 4 Stacker crane TW-ATSP instances

Instance	$\lvert I \rvert$	Opt value	Pro-CG					[2]
			No. of iterations	No. of cuts	Percentage of gap closed	Percentage of time to get the bound	Percentage of final gap	Percentage of final gap
rbg010a	12	149	227	526	≥ 99.07	7.50	0.67	0.67
rbg017	17	148	255	793	≥ 78.07	27.89	14.86	0
rbg017.2	17	107	199	504	≥ 96.90	27.80	1.87	0
rbg016a	18	179	422	1,505	≥ 97.08	100.00	1.68	1.11
rbg016b	18	142	245	632	≥ 86.54	31.77	10.56	6.33
rbg017a	19	146	219	636	≥ 95.07	39.42	2.74	0
rbg019a	21	217	552	1,962	≥ 97.40	100.00	1.38	0
rbg019b	21	182	675	1,697	≥ 89.47	100.00	6.59	1.09
rbg019c	21	190	258	792	≥ 70.21	23.12	20.53	4.21
rbg019d	21	344	608	1,776	≥ 90.57	100.00	4.94	0.29
rbg021	21	190	257	633	≥ 72.05	20.27	20.53	4.21
rbg021.2	21	182	300	692	≥ 77.00	25.32	17.03	0
rbg021.3	21	182	487	1,348	≥ 74.40	100.00	19.23	2.19
rbg021.4	21	179	416	1,134	≥ 76.66	77.66	17.88	1.11
rbg021.5	21	169	306	908	≥ 77.67	81.93	17.16	1.18
rbg021.6	21	134	294	743	≥ 96.60	58.94	2.24	0.74
rbg021.7	21	133	263	658	≥ 95.64	53.51	3.01	3.75
rbg021.8	21	132	346	744	≥ 96.12	36.53	3.03	2.27
rbg021.9	21	132	369	761	≥ 95.18	56.28	3.79	3.03
rbg020a	22	210	399	1,150	≥ 77.95	100.00	14.29	0
rbg027a	29	268	667	1,655	≥ 76.11	100.00	16.04	0.74

different from those that are currently used in MILP solvers and that it is worth exploring heuristics that generate them more efficiently.

A second set of experiments has been performed to test the effectiveness of pro-CG cuts in the context of the simple model for the TW-ATSP discussed in Sect. 4, where the basic ILP model (22) only includes in- and out-degree equations (no subtour elimination constraints are exploited). Note that no continuous variables are present in the objective function of this model. Table 4 reports results on TW-ATSP real-world instances introduced by Ascheuer [1], derived "from an industry project with the aim to minimize the unloaded travel time of a stacker crane within an automated storage system".

In particular, we report results on a set of 21 problems of small/medium size, with up to 30 vertices. The information provided in Table 4 for each instance is the number of cities ($|I|$) and the optimal solution value (opt value). For pro-CG separation, Table 4 gives the same information as in Table 1. As a comparison, we provide the final gap obtained by the pro-CG closure and the gap at the root node in [2] (note that we report the gap instead of the gap closed because a different initial formulation is used in [2]). Computing time is also not reported since the 1,200-second time limit is reached for all our TW-ATSP instances. Instead, we report the percentage of time (with respect to the time limit) spent to find the final bound. For example, for problem `rbg010a` the algorithm improves the bound from 42 to 148 in 90.1 CPU seconds (7.50% of the total time) and spends the remaining computing time without finding any new cut. In such a case, we may guess that we are close to the Chvátal–Gomory closure, but proving that no violated pro-CG cut exists can require a great deal of enumeration.

The results for TW-ATSP instances are also very encouraging. Although our initial model is known to be very weak, pro-CG cuts are able to close a very significant amount (always more than 70%) of the initial gap. This suggests that pro-CG cuts could be used successfully together with special purpose (polyhedral) separation routines in an attempt to improve the overall behavior of a cutting plane algorithm.

6 Conclusions

In this paper we have introduced a projected version of the classical Chvátal–Gomory cuts, and have studied their practical effectiveness for MIPLIB instances and for some special classes of MILP problems. Our approach is to project first the linear programming relaxation of the MILP at hand onto the space of the integer variables, and then to derive CG cuts for the projected polyhedron.

Although there are cases where they are ineffective, projected CG cuts provide excellent bounds for a number of MIPLIB instances. Furthermore, they can be applied successfully on a wide range of combinatorial problems where the continuous variables do not appear in the objective function. Our experiments on TW-ATSP confirm this claim–even starting from a very weak formulation involving big-M coefficients, the use of projected CG cuts is able to close a large portion of the integrality gap (70% or more, in our test cases). In our view, these results give a concrete hope that a similar performance can be obtained on other classes of problems (including scheduling and cutting/packing problems) when they are modeled through weak formulations involving continuous variables linked to the integer ones by constraints involving big-M coefficients.

A natural question (and one posed by the editor and referees) is whether using the multipliers u in a solution of (14)–(19) to write GMI cuts instead of pro-CG cuts results in a substantial improvement. After all, GMI cuts dominate pro-CG cuts. In our limited experiments with MIPLIB instances, we did not notice much of an improvement, but we do not claim to have a conclusive answer.

Acknowledgments We warmly thank Jon Lee and François Margot for friendly and useful discussions on the subject. We also thank the referees for useful comments and suggestions. Part of this research has been carried out when the last author was Herman Goldstine Fellow of the IBM T.J. Watson Research Center whose support is strongly acknowledged.

References

1. Ascheuer, N.: Hamiltonian path problems in the on-line optimization of flexible manufacturing systems. PhD Thesis, Technische Universität Berlin, Berlin (1995)
2. Ascheuer, N., Fischetti, M., Grötschel, M.: A polyhedral study of the asymmetric travelling salesman problem with time windows. Networks **36**, 69–79 (2000)

3. Balas, E., Ceria, S., Cornuéjols, G.: A lift-and-project cutting plane algorithm for mixed 0-1 programs. Math. Program. **58**, 295–324 (1993)
4. Balas, E., Saxena, A.: Optimizing over the Split Closure: Modeling and Theoretical Analysis, IMA "Hot Topics" Workshop: Mixed Integer Programming, Minneapolis, 25–29 July 2005
5. Bixby, R.E., Ceria, S., McZeal, C.M., Savelsbergh, M.W.P.: MIPLIB 3.0, http://www.caam.rice.edu/~bixby/miplib/miplib.html
6. Bonami, P., Minoux, M.: Using rank-1 lift-and-project closures to generate cuts for 0-1 MIPs, a computational investigation. Discrete Optim. **2**, 288–307 (2005)
7. Caprara, A., Letchford, A.N.: On the separation of split cuts and related inequalities. Math. Program. **94**, 279–294 (2003)
8. Chvátal, V.: Edmonds polytopes and a hierarchy of combinatorial problems C73. Discrete Math. **4**, 305–337 (1973)
9. Codato, G., Fischetti, M.: Combinatorial Benders' cuts for mixed-integer linear programming. Oper. Res. **54**, 756–766 (2006)
10. COIN-OR. www.coin-or.org
11. Cook, W., Kannan, R., Schrijver, A.: Chvátal closures for mixed integer programming problems. Math. Program. **47**, 155–174 (1990)
12. Cornuéjols, G., Li, Y.: On the rank of mixed 0,1 polyhedra. Math. Program. **91**, 391–397 (2002)
13. Cornuéjols, G., Li, Y.: A connection between cutting plane theory and the geometry of numbers. Math. Program. **93**, 123–127 (2002)
14. Dash, S., Günlük, O., Lodi, A.: On the MIR closure of polyhedra. IBM, T.J. Watson Research, Working paper (2005)
15. Eisenbrand, F.: On the membership problem for the elementary closure of a polyhedron. Combinatorica **19**, 297–300 (1999)
16. Fischetti, M., Lodi, A. : Optimizing over the first Ch closure. In: Jünger, M., Kaibel, V. (eds.) Integer Programming and Combinatorial Optimization—IPCO 2005, LNCS 3509., pp. 12–22. Springer, Berlin Heidelberg New York (2005)
17. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bull AMS **64**, 275–278 (1958)
18. Gomory, R.E.: An algorithm for integer solutions to linear programs. In: Graves, R.L., Wolfe, P. (eds.) Recent Advances in Mathematical Programming., pp. 269–302. McGraw-Hill, New York (1963)
19. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin Heidelberg New York (1988)
20. ILOG Cplex 9.0: User's Manual and Reference Manual, ILOG, S.A., http://www.ilog.com/ (2005)
21. Klau, G.W., Mützel, P.: Optimal labelling of point features in rectangular labelling models. Math. Program. **94**, 435–458 (2003)
22. Marchand, H., Wolsey, L.A.: Aggregation and mixed integer rounding to solve MIPs. Oper. Res. **49**, 363–371 (2001)
23. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)
24. Nemhauser, G.L., Wolsey, L.A.: A recursive procedure to generate all cuts for 0-1 mixed integer programs. Math. Program. **46**, 379–390 (1990)
25. Schrijver, A.: On cutting planes. Ann. Discrete Math. **9**, 291–296 (1980)