J Glob Optim (2018) 70:289–306
https://doi.org/10.1007/s10898-017-0562-5

Generalized coefficient strengthening cuts for mixed integer programming

Wei-Kun Chen[1] · Liang Chen[1] · Mu-Ming Yang[1] · Yu-Hong Dai[1]

Received: 8 March 2017 / Accepted: 24 August 2017 / Published online: 2 September 2017

Abstract Cutting plane methods are an important component in solving the mixed integer programming (MIP). By carefully studying the coefficient strengthening method, which is originally a presolving method, we are able to generalize this method to generate a family of valid inequalities called generalized coefficient strengthening (GCS) inequalities. The invariant property of the GCS inequalities is established under bound substitutions. Furthermore, we develop a separation algorithm for finding the violated GCS inequalities for a general mixed integer set. The separation algorithm is proved to have the polynomial time complexity. Extensive numerical experiments are made on standard MIP test sets, which demonstrate the usefulness of the resulting GCS separator.

Keywords Mixed integer programming · Cutting plane method · Separation algorithm · Coefficient strengthening

This work was supported by the Chinese Natural Science Foundation (Nos. 11631013, 71331001, 11331012) and the National 973 Program of China (No. 2015CB856002).

✉ Mu-Ming Yang
ymm@lsec.cc.ac.cn

Wei-Kun Chen
cwk@lsec.cc.ac.cn

Liang Chen
chenliang15@lsec.cc.ac.cn

Yu-Hong Dai
dyh@lsec.cc.ac.cn

[1] Institute of Computational Mathematics and Scientific/Engineering Computing, State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

1 Introduction

Consider the mixed integer programming (MIP) of the form

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax \leq \beta \\ & l \leq x \leq u \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{aligned} \tag{1}$$

where $A \in \mathbb{Q}^{m \times n}$ is a rational matrix, $c \in \mathbb{Q}^n$, $\beta \in \mathbb{Q}^m$ and $l,\ u \in \mathbb{Z}^p \times \mathbb{Q}^{n-p}$.

Cutting plane methods, which were proposed in [1] more than 50 years ago, have become one of the indispensable ingredients for solving MIP. As they can automatically improve the formulation of the original problem, cutting plane methods are contained in state of the art MIP solvers. General cutting planes, which do not require any knowledge about the problem structure, include Gomory Mixed Integer (GMI) Cuts [1], Mixed Integer Rounding (MIR) Cuts [2], Strong Chvátal–Gomory Cuts [3], Zero-half Cuts [4], Implied Bound Cuts [5], Clique Cuts [6,7] and Disjunctive Cuts [8]. For special cutting planes, which make use of the problem structure, we have Knapsack Cover Cuts [6,9], Flow Cover Cuts [11,12], Flow Path Cuts [10], GUB Cover Cuts [13], Multi-commodity Flow Cuts [14], etc. The article [15] studied the impact of cutting plane methods in the commercial MIP solver CPLEX [16] and a 6.1× speedup is concluded on the set of test instances which required at least 10 s to solve by at least one version of the solvers.

Another key ingredient for solving MIP is the presolving method. It contains a set of routines that reduce the size of problem and strengthen the formulation of a given model. See [17] for a good survey about presolving methods. Here we just mention that the coefficient strengthening method [6,18,19] is one of the presolving methods and attempts to repair a bad formulation of the original problem (see also Sect. 2).

Although the coefficient strengthening method is a presolving method, we are able to generalize this method to generate the so-called generalized coefficient strengthening (GCS) inequality. Specifically, the GCS inequality is derived based on the general mixed integer set

$$M_I = M \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p}), \tag{2}$$

where M is the relaxation set

$$M = \{x \in \mathbb{R}^n : a^T x \leq b,\ l \leq x \leq u\}. \tag{3}$$

Here $a^T = (a_1, a_2, a_3, \ldots, a_n)$ is any row of A and b is the corresponding right hand side. We show that GCS inequalities are invariant under bound substitutions. Furthermore, by analyzing variables bounds and utilizing the violated information, we are able to generalize the idea of [20] to develop a separation algorithm for finding GCS inequalities for the general mixed integer set M_I. The separation algorithm is proved to have the polynomial time complexity. Our numerical experiments demonstrated the usefulness of the resulting GCS separator.

Here we should notice that in [21], the authors considered a special case of M_I, that is

$$M_I^{(0)} = \{(x, y) \in \mathbb{R}^n \times \mathbb{Z} : \sum_{i=1}^{n} x_i \leq Dy,\ 0 \leq x \leq u,\ y \geq 0\}, \tag{4}$$

where $D > 0$. They studied the convex hull of $M_I^{(0)}$ and derived an exponential family of valid inequalities, called residual capacity inequalities. It will be shown in Sect. 3 that such residual capacity inequalities are special GCS inequalities. Furthermore, a linear time separation algorithm is presented in [20] for finding the violated residual capacity inequalities. The separation algorithm proposed in this paper is based on [20], but can handle arbitrary variables bounds and arbitrary coefficients.

This paper is arranged as follows. In Sect. 2, we review the coefficient strengthening method. In Sect. 3, we extend the idea of the coefficient strengthening method by using more redundant information of M_I, which leads to the family of GCS inequalities. We show that GCS inequalities are invariant under bound substitutions. In Sect. 4, we first discuss the importance of infinite bounds for generating GCS inequalities. Then a polynomial time separation algorithm is presented to find the violated cuts. In Sect. 5, we test the effectiveness of the GCS cuts by inserting it as a separator in the open source solver SCIP [22]. Computational results on the test sets MIPLIB 3.0 [23], 2003 [24] and 2010 [25] are presented. Finally, in Sect. 6, we draw some conclusions and give some directions of future research.

Throughout this paper, we assume that the data is rational. Without loss of generality, we assume that $a_i \neq 0$ for all $1 \leq i \leq n$ since if $a_i = 0$, x_i can be dropped. Let $N := \{1, 2, 3, \cdots, n\}$ and $I := \{1, 2, 3, \ldots, p\}$ be the index set of all variables and integer variables, respectively. We denote $\overline{\alpha}$ be the *maximal activity* of $a^T x$ which defined as $\overline{\alpha} := \max\{a^T x : l \leq x \leq u\} = \sum_{i:a_i>0} a_i u_i + \sum_{i:a_i<0} a_i l_i$. The inequality $\pi^T x \leq \pi_0$ is called a *valid* inequality for M_I if it is satisfied by all points in M_I. Given any valid inequality $\pi^T x \leq \pi_0$ for M_I, we say $\pi^T x \leq \pi_0$ *dominates* $a^T x \leq b$ on M if $M' := \{x \in \mathbb{R}^n : \pi^T x \leq \pi_0,\ l \leq x \leq u\} \subseteq M$. Moreover, if $M' \subsetneq M$, then $\pi^T x \leq \pi_0$ *strictly dominates* $a^T x \leq b$ on M.

2 Coefficient strengthening method

In this section, we review the coefficient strengthening method [6,18,19]. The basic idea of the coefficient strengthening method is to change the coefficient of one integer variable and the right hand side via the redundant information so that the relaxation set M [as defined in (3)] is tighter without affecting the mixed integer set M_I. In this case, the constraint will become redundant if the corresponding variable does not take the value of the lower or upper bound. The details of the method is described in the following proposition. A proof is presented for completeness.

Proposition 1 *Let M_I be the mixed integer set in (2). For $a_j > 0,\ j \in I$, if $0 < \overline{\alpha} - b < a_j$, then $a'^T x \leq b'$ is valid for M_I, where*

$$\begin{cases} a'_i = a_i, & i \neq j, \\ a'_j = \overline{\alpha} - b, \\ b' = b - (a_j - a'_j)u_j. \end{cases}$$

Furthermore, $a'^T x \leq b'$ strictly dominates $a^T x \leq b$ on M.

Proof Notice that $0 < a'_j < a_j$. Given any point $x' \in M_I$, consider the following two cases.

(i) $x'_j \in M_I \cap \{x \in \mathbb{R}^n : x_j \leq u_j - 1\}$. In this case, we have that

$$\begin{aligned} a'^T x' &\leq \overline{\alpha} - a_j u_j + a'_j x'_j \leq \overline{\alpha} - a_j u_j + a'_j (u_j - 1) \\ &= \overline{\alpha} - (a_j - a'_j) u_j + b - \overline{\alpha} = b', \end{aligned}$$

where the first inequality follows from the definition of $\overline{\alpha}=\sum_{i:\,a_i>0} a_i u_i + \sum_{i:\,a_i<0} a_i l_i$.

(ii) $x'_j \in M_I \cap \{x \in \mathbb{R}^n : x_j = u_j\}$. In this case, it follows that

$$a'^T x' = a^T x' + (a'_j - a_j)x_j \le b + (a'_j - a_j)u_j = b'.$$

Combining the above, we know that $a'^T x \le b'$ is valid for M_I.

For the second part, we first prove that $\{x \in \mathbb{R}^n : a'^T x \le b',\ l \le x \le u\} \subseteq M$. In fact, for any point x' with $a'^T x' \le b'$ and $l \le x' \le u$, it follows that

$$a^T x' = a'^T x' + (a_j - a'_j)x'_j \le b' + (a_j - a'_j)u_j = b.$$

It remains to show that there exists x' with $l \le x' \le u$ such that $a^T x' \le b$ and $a'^T x' > b'$. Pick x' with $x'_i = u_i$ if $a_i > 0$, $x'_i = l_i$ if $a_i < 0$ for all $i \ne j$ and $u_j - 1 < x'_j < \frac{b-\overline{\alpha}+a_j u_j}{a_j}$. It follows that

$$a^T x' = \overline{\alpha} - a_j u_j + a_j x'_j \le \overline{\alpha} - a_j u_j + a_j \frac{b - \overline{\alpha} + a_j u_j}{a_j} = b$$

and

$$a'^T x' = \overline{\alpha} - a_j u_j + a'_j x'_j > \overline{\alpha} - a_j u_j + a'_j (u_j - 1) = b - (a_j - a'_j)u_j = b'.$$

Therefore, $a'^T x \le b'$ strictly dominates $a^T x \le b$ on M. □

Remark 1 Consider $n = 2$ and assume that $I = \{1\}$. If $0 < \overline{\alpha} - b < a_1$, it is not difficult to verify that the two points $(u_1 - 1, \frac{\overline{\alpha}-a_1 u_1}{a_2})$ and $(u_1, \frac{b-a_1 u_1}{a_2})$ are extreme points of $\mathrm{Conv}(M_I)$. Combining the two extreme points gives the valid inequality $a'_1 x_1 + a_2 x_2 \le b'$, where $a'_1 = \overline{\alpha} - b$ and $b' = b - (a_1 - a'_1)u_1$.

From the above proof, we know that if $0 < \overline{\alpha} - b < a_j$, $a^T x \le b$ will be satisfied when $x_j \le u_j - 1$. This is also the reason why the coefficient a_j can be strengthened.

By Proposition 1, we know that under suitable assumptions, a tighter relaxation set can be obtained without affecting the mixed integer set M_I by replacing $a^T x \le b$ with $a'^T x \le b'$. This can also be interpreted as modifying one coefficient a_j and the right hand side b in the original constraint that $a^T x \le b$. A similar conclusion can be drawn for the case that $a_j < 0$, $j \in I$. Thus all coefficients of integer variables are likely to be strengthened to tighten the relaxation set.

3 Generalized coefficient strengthening inequality

The condition that $0 < \overline{\alpha} - b < a_j$ is quite strong as it requires that x_j plays a significant role in the inequality that $a^T x \le b$. This restricts the applications of the coefficient strengthening method for solving MIP [17]. Assume again that $a_j > 0$ for some $j \in I$. To weaken the condition, we may look for some integer $r \in \{0, 1, 2, \ldots, u_j - l_j - 1\}$ such that $ra_j < \overline{\alpha} - b < (r + 1)a_j$. Then we can see that the inequality that $a^T x \le b$ is redundant for $x_j \le u_j - r - 1$. Motivated by this observation, we will derive a family of valid inequalities called generalized coefficient strengthening (GCS) inequalities. Moreover, we will provide the invariant features of the GCS inequalities. As will be seen, although this generalization of the coefficient strengthening method cannot act as a presolving method any more, the GCS inequalities can be used as cutting planes to cut off the current relaxation solution.

Proposition 2 *Assume that $a_j > 0$ for some $j \in I$. If there exists some integer $r \in \{0, 1, 2, \ldots, u_j - l_j - 1\}$ such that $ra_j < \overline{\alpha} - b < (r+1)a_j$, then*

(1) $a'^T x \le b'$ is valid for the mixed integer set M_I in (2), where

$$\begin{cases} a'_i = a_i, \qquad i \ne j, \\ a'_j = \overline{\alpha} - b - ra_j, \\ b' = b - (a_j - a'_j)(u_j - r); \end{cases}$$

(2) $a'^T x \le b'$ dominates $a^T x \le b$ on $M \cap \{x \in \mathbb{R}^n : x_j \le u_j - r\}$ and $a^T x \le b$ dominates $a'^T x \le b'$ on $M \cap \{x \in \mathbb{R}^n : x_j \ge u_j - r\}$.

Proof Let $x' \in M_I \cap \{x \in \mathbb{R}^n : x_j \le u_j - r - 1\}$. It follows from $a'_j > 0$ and the defintion of $\overline{\alpha}$ that

$$a'^T x' \le \overline{\alpha} - a_j u_j + a'_j x'_j \le \overline{\alpha} - a_j u_j + a'_j (u_j - r - 1) = b'.$$

In case of $x'_j \in M_I \cap \{x \in \mathbb{R}^n : x_j \ge u_j - r\}$, we also have that

$$a'^T x' = a^T x' - a_j x'_j + a'_j x'_j \le b - (a_j - a'_j)(u_j - r) = b'.$$

Thus $a'^T x' \le b'$ is valid for M_I.

For the second part, we may just consider the case $x_j \le u_j - r$, as the other case is similar. In this case, for any point x' with $l \le x' \le u$ and $a'^T x' \le b'$, it follows from $a'_j < a_j$ that

$$a^T x' = a'^T x' + (a_j - a'_j)x'_j \le b' + (a_j - a'_j)(u_j - r) = b.$$

This completes the proof. □

Remark 2 Consider the disjunction $M_1 := M_I \cap \{x \in \mathbb{R}^n : x_j \le u_j - r - 1\}$ and $M_2 := M_I \cap \{x \in \mathbb{R}^n : x_j \ge u_j - r\}$. In the proof of Proposition 2, we prove the validity of the inequality $a'^T x \le b'$ by verifying that it is valid for M_1 and M_2. Therefore, $a'^T x \le b'$ may be viewed as a special disjunctive cut [8].

Similarly, we have the following result for the case that $a_j < 0$ for $j \in I$.

Proposition 3 *Assume that $a_j < 0$ for some $j \in I$. If there exists $r \in \{0, 1, 2, \ldots, u_j - l_j - 1\}$ such that $-ra_j < \overline{\alpha} - b < -(r+1)a_j$, then*

(1) $a'^T x \le b'$ is valid for the mixed integer set M_I in (2), where

$$\begin{cases} a'_i = a_i, \qquad i \ne j, \\ a'_j = -(\overline{\alpha} - b + ra_j), \\ b' = b - (a_j - a'_j)(l_j + r), \end{cases}$$

(2) $a'^T x \le b'$ dominates $a^T x \le b$ on $M \cap \{x : x_j \ge l_j + r\}$ and $a^T x \le b$ dominates $a'^T x \le b'$ on $M \cap \{x : x_j \le l_j + r\}$.

For convenience, we call the valid inequality derived in Propositions 2 or 3 generalized coefficient strengthening (GCS) inequality. Notice that in the case that $r = 0$, Proposition 2 reduces to Proposition 1 since $M \cap \{x \in \mathbb{R}^n : x_j \le u_j - r\} = M$. Comparing with Proposition 1, the condition that $ra_j < \overline{\alpha} - b < (r+1)a_j$ for some $r \in \{0, 1, 2, \ldots, u_j - l_j - 1\}$ in Proposition 2 is more general, but its conclusion is not so strong as that in Proposition 1. As illustrated in the following example, although the GCS inequality is still valid for M_I, we can not make a simple substitution to the original constraint that $a^T x \le b$.

Example 1 Consider the mixed integer set

$$M_I^{(1)} = \{(x_1, x_2) \in \mathbb{Z}^2 : 2.7x_1 + 2.2x_2 \leq 11,\ 0 \leq x_1 \leq 3,\ 0 \leq x_2 \leq 4\}.$$

It is easy to see that the maximal activity of $a^T x$ over the set $M_I^{(1)}$ is $\overline{\alpha} = 16.9$. For variable x_1, since $2 \times a_1 < \overline{\alpha} - b = 5.9 < 3 \times a_1$, let $r = 2$. Take $a'_1 = 5.9 - 2 \times 2.7 = 0.5$ and $b' = 11 - (2.7 - 0.5) \times (3 - 2) = 8.8$. By Proposition 2, we know that the inequality

$$0.5x_1 + 2.2x_2 \leq 8.8$$

is valid for $M_I^{(1)}$. However, simply replacing $2.7x_1 + 2.2x_2 \leq 11$ with $0.5x_1 + 2.2x_2 \leq 8.8$ in $M_I^{(1)}$ brings an extra integer point (3, 3), which is not feasible to the original set $M_I^{(1)}$.

Nevertheless, we may add the GCS inequality to the MIP formulation to tighten the relaxation set.

Proposition 4 *Consider the GCS inequality derived in Proposition 2. There must exist some $x' \in M$ such that $a^T x' \leq b$ but $a'^T x' > b'$. Thus, adding $a'^T x \leq b'$ into the constraints of M_I leads to a more compact relaxation set.*

Proof Take any x' satisfying

$$\sum_{i \neq j} a_i x'_i = \overline{\alpha} - a_j u_j \quad \text{and} \quad x'_j \in (u_j - r - 1, \frac{b - \overline{\alpha} + a_j u_j}{a_j}].$$

By direct calculations, we know that

$$a^T x' = \overline{\alpha} - a_j u_j + a_j x'_j \leq \overline{\alpha} - a_j u_j + a_j \frac{b - \overline{\alpha} + a_j u_j}{a_j} = b$$

and

$$\begin{aligned} a'^T x' &= \overline{\alpha} - a_j u_j + a'_j x'_j > \overline{\alpha} - a_j u_j + a'_j (u_j - r - 1) \\ &= \overline{\alpha} - a_j u_j + a'_j (u_j - r) - \overline{\alpha} + b + r a_j = b'. \end{aligned}$$

This completes our proof. □

Therefore, although the above generalization of the coefficient strengthening method cannot be used as a presolving method any more as the purpose of presolving method is to reduce the size of the original problem, we may add the new valid inequality to the MIP formulation as a cutting plane to tighten the relaxation set. This will also be helpful to solve MIP as demonstrated by our numerical experiments in Sect. 5.

In what follows, we describe the invariant property of the GCS inequality. For any variable x_k, it is shown that the GCS inequality is invariant under either of the bound substitutions

$$y_k := x_k - l_k \text{ and } y_k := u_k - x_k. \tag{5}$$

Proposition 5 *Assume that M_I is the mixed integer set in (2). For any $k \in N$, let $y_k := x_k - l_k$ and $y_i := x_i,\ i \neq k$ and consider the corresponding transformed mixed integer set*

$$M_I^y := \{y \in \mathbb{Z}^p \times \mathbb{R}^{n-p} :\ a^T y \leq b - a_k l_k,\ l^y \leq y \leq u^y\},$$

where $l_k^y = 0$, $u_k^y = u_k - l_k$ and $l_i^y = l_i$, $u_i^y = u_i$, $i \neq k$. If there exists a GCS inequality that $a'^T x \leq b'$ for M_I given by Propositions 2 or 3, then there also exists a GCS inequality for M_I^y which is equivalent to $a'^T x \leq b'$.

Proof We may just consider the case that $a'^T x \le b'$ is generated by Proposition 2 as the other case is similar. The definition of M_I^y indicates that $\overline{\alpha}^y = \overline{\alpha} - a_k l_k$ and $\overline{\alpha}^y - (b - a_k l_k) = \overline{\alpha} - b$. By Proposition 2, $a'^T y \le b - a_k l_k - (a_j - a'_j)(u_j^y - r)$ is valid for M_I^y. Now consider the following two cases.

(i) If $k \ne j$, then $u_j^y = u_j$.

(ii) If $k = j$, then $u_j^y = u_j - l_j$.

In either case, the substitution that $y_k = x_k - l_k$ and $y_i = x_i,\ i \ne k$ leads to the same GCS inequality $a'^T x \le b'$. □

Similarly, we have the following proposition.

Proposition 6 *Assume that M_I is the mixed integer set in* (2). *For any $k \in N$, let $y_k := u_k - x_k$ and $y_i := x_i,\ i \ne k$ and consider the corresponding transformed mixed integer set*

$$M_I^y := \{y \in \mathbb{Z}^p \times \mathbb{R}^{n-p} :\ (a^y)^T y \le b - a_k u_k, l^y \le y \le u^y\},$$

where $a_k^y = -a_k$, $l_k^y = 0$, $u_k^y = u_k - l_k$ and $a_i^y = a_i$, $l_i^y = l_i$, $u_i^y = u_i$, $i \ne k$. If there exists a GCS inequality that $a'^T x \le b'$ for M_I given by Propositions 2 *or* 3, *then there also exists a GCS inequality for M_I^y which is equivalent to $a'^T x \le b'$.*

Now we shall mention more possibilities of generating the GCS inequalities. At first, for the same constraint, different GCS inequalities may be generated if different integer variables are chosen. Consider Example 1 again. It is easy to verify that, if we pick the integer variable x_2, the GCS inequality that $2.7x_1 + 1.5x_2 \le 9.6$ will be generated, which is valid for X as well. In addition, we may also drop some terms of the constraint to derive GCS inequalities, as illustrated in the following example.

Example 2 Consider the mixed integer set

$$M_I^{(2)} = \{(x_1, x_2, x_3) \in \mathbb{Z}^3 : 2.7x_1 + 2.2x_2 + 2.2x_3 \le 11,\ 0 \le x_1 \le 3,\ 0 \le x_2 \le 4,\ 0 \le x_3 \le 4\}.$$

If dropping the term $2.2x_3$ from the linear inequality of $M_I^{(2)}$, we are led to the reduced inequality that $2.7x_1 + 2.2x_2 \le 11$, which is valid for $M_I^{(2)}$ due to $2.2x_3 \ge 0$. Furthermore, by Example 1, we can generate the GCS inequality that $0.5x_1 + 2.2x_2 \le 8.8$ based on the reduced mixed integer set, which is exactly $M_I^{(1)}$ in Example 1. This GCS inequality is valid for $M_I^{(2)}$ as well. In a similar way, dropping the term $2.2x_2$ from the linear inequality of $M_I^{(2)}$ can yield another GCS inequality that $0.5x_1 + 2.2x_3 \le 8.8$.

As a matter of fact, we can also consider to drop multiple terms of the constraint to generate GCS inequalities. As shown in the next section, this approach is very important for how to make better use of the GCS technique in solving MIP. For convenience, for $i \in N$, we call the term $a_i x_i$ is an *inactive term* if it is dropped from the inequality and the corresponding variable x_i is called an *inactive variable*. The term $a_i x_i$ which remains in the inequality is called an *active term* and the corresponding variable x_i is called an *active variable*. In addition, for $j \in I$, we call the integer variable x_j by *GCS variable* if we are going to consider the GCS inequality based on this variable. All GCS inequalities generated by choosing different GCS variables and different active variables constitute the family of GCS inequalities.

To end this section, we remark that the residual capacity inequality [21] for the set $M_I^{(0)}$ in (4) is a GCS inequality.

Proposition 7 *Consider the mixed integer set $M_I^{(0)}$ in (4). Let $S \subseteq N$, $\eta = \lceil \frac{\sum_{i \in S} u_i}{D} \rceil$ and $\tau = \sum_{i \in S} u_i - D(\eta - 1)$. If $0 < \tau < D$, then the residual capacity inequality*

$$\sum_{i \in S} x_i \leq \tau y + (\eta - 1)(D - \tau) \tag{6}$$

is a GCS inequality.

Proof Consider the inequality with active variables being y and x_i, $i \in S$,

$$\sum_{i \in S} x_i \leq Dy. \tag{7}$$

The maximal activity of (7) over $M_I^{(0)}$ is $\overline{\alpha} = \sum_{i \in S} u_i$. Since $0 < \tau < D$,

$$-(-D)(\eta - 1) < \sum_{i \in S} u_i - 0 < -(-D)\eta.$$

Choosing y to be the GCS variable, Proposition 3 gives the following GCS inequality

$$\sum_{i \in S} x_i - \tau y \leq -(-D + \tau)(0 + \eta - 1),$$

which is exactly the residual capacity inequality (6). □

4 Separation algorithm for finding violated GCS inequalities

As there are usually many GCS inequalities for a mixed integer set, like other cutting planes, only some of them can be allowed to add into the problem. Therefore, we have to develop an efficient strategy for trying to generate the most effective ones. The basic thing is that, given any LP relaxation x^*, we attempt to find some GCS inequalities which cut off x^*. After some considerations about the importance of infinite bounds on variables for generating GCS inequalities, a mixed integer nonlinear programming model is formulated for the separation problem. By using the violated information, a separation algorithm is designed for solving the separation problem. It is proved that the separation algorithm is a polynomial time algorithm.

4.1 Variables bounds on generating GCS inequalities

As mentioned in Sect. 3, different GCS inequalities can be generated by choosing different active variables. In the following, we shall consider which variables must be active or not. In order to keep the validity of the inequality, we may adopt either of the following bound substitutions

$$y_i := x_i - l_i \quad \text{and} \quad y_i := u_i - x_i.$$

to disregard the term $a_i x_i$ by dropping $a_i y_i$ or $-a_i y_i$. A key observation is that those variables with infinite bounds play an important role in generating GCS inequalities.

Proposition 8 *Consider the mixed integer set M_I in (2) and any variable x_i.*

(1) If $a_i > 0$, $l_i \neq -\infty$, $u_i = \infty$ or $a_i < 0$, $l_i = -\infty$, $u_i \neq \infty$, no GCS inequalities can be generated unless x_i is an inactive variable;
(2) If $a_i > 0$, $l_i = -\infty$, $u_i \neq \infty$ or $a_i < 0$, $l_i \neq -\infty$, $u_i = \infty$, x_i should be active in every GCS inequality;
(3) If $l_i = -\infty$ and $u_i = \infty$, no GCS inequalities can be generated.

Proof Consider the case that $a_i > 0$, $l_i \neq -\infty$, $u_i = \infty$. If x_i is an active variable, then $\overline{\alpha} = \infty$, which implies that $\overline{\alpha} - b > (r+1)a_i$ for any r. Therefore, no GCS inequalities can be generated unless x_i is an inactive variable. The proof is similar for the case that $a_i < 0$, $l_i = -\infty$, $u_i \neq \infty$.

For the second part, we may only consider the case that $a_i > 0$, $l_i = -\infty$, $u_i \neq \infty$. As $l_i = \infty$ and $a_i(x_i - u_i) \leq 0$, in order to keep the validity of the inequality, x_i should be active in every GCS inequality.

Combining the above, if x_i is a free variable, the maximal activity $\overline{\alpha}$ must be ∞ and x_i should be active in every GCS inequality. Thus, no GCS inequalities can be generated. □

Due to Proposition 8, the remaining case is that $l_i \neq -\infty$ and $u_i \neq \infty$. In fact, the variable x_i could be either active or inactive in this case. More exactly, x_i may stay in the constraint as an active variable or become an inactive variable by dropping $a_i(x_i - l_i)$ if $a_i > 0$ or $-a_i(u_i - x_i)$ if $a_i < 0$.

Furthermore, by Propositions 5 and 6, the GCS inequality is invariant under bound substitutions. Therefore, in the remainder of this paper, we assume that for any variable x_i,

$$\begin{aligned} l_i = 0,\ u_i < \infty, \quad \text{if } a_i > 0; \\ l_i = 0,\ u_i = \infty, \quad \text{if } a_i < 0. \end{aligned} \tag{8}$$

Under the assumption (8), disregarding the term $a_i x_i$ with $a_i > 0$ does not affect the right hand side b and the term $a_i x_i$ with $a_i < 0$ must be an active term.

4.2 An MINLP model for the separation problem

To begin with, we shall analyze the effect of disregarding the term $a_i x_i$ for any variable x_i with $a_i > 0$. On one hand, observe that the difference between the new maximal activity $\overline{\alpha}'$ and the right hand side decreases, since

$$\overline{\alpha}' - b = \overline{\alpha} - a_i u_i - b < \overline{\alpha} - b.$$

This is beneficial to find a suitable r such that $ra_j < \overline{\alpha} - b < (r+1)a_j$. On the other hand, consider the current relaxation x^* with the activity $t^* = a^T x^*$. After disregarding the term $a_i x_i$, the difference between the new activity t'^* and the right hand side is decreasing as well, since we have

$$t'^* - b = t^* - a_i x_i^* - b \leq t^* - b.$$

This is not in accordance with our purpose, which aims to find a violated inequality. Therefore, it is difficult to choose those active terms directly.

As mentioned in Sect. 1, a linear time separation algorithm is proposed in [20] for the special mixed integer set $M_I^{(0)}$. With the help of the analysis of variables bounds in Proposition 8 and the invariant property of the GCS inequality under bound substitutions in Propositions 5 and 6, we are able to adapt the algorithm in [20] to develop a separation algorithm for finding GCS inequalities for the general mixed integer set M_I.

Assume that M_I is the normalized mixed integer set satisfying (8) and denote x^* to be the current relaxation point. Then by Proposition 8, we know that all variable x_i should be active in every GCS inequality for $i \in K_2 := \{i \in N : a_i < 0\}$. Fixing any GCS variable x_j with $a_j > 0$, we shall ask whether it is possible to choose an integer value of r and the index set of active variables, $S \subseteq K_1 := \{i \in N : a_i > 0, i \neq j\}$, such that

$$a'_j x^*_j + \sum_{i\in S} a_i x^*_i + \sum_{i\in K_2} a_i x^*_i > b', \tag{9}$$

where

$$a'_j = \sum_{i\in S} a_i u_i + a_j u_j - b - r a_j \quad \text{and} \quad b' = b - (a_j - a'_j)(u_j - r).$$

This is a combinatorial optimization problem. By introducing binary variables, such a problem can be reformulated to look for suitable $z_i \in \{0, 1\}$ for $i \in K_1$ such that

$$a'_j x^*_j + \sum_{i\in K_1} a_i x^*_i z_i + \sum_{i\in K_2} a_i x^*_i > b', \tag{10}$$

where

$$a'_j = \sum_{i\in K_1} a_i u_i z_i + a_j u_j - b - r a_j \quad \text{and} \quad b' = b - (a_j - a'_j)(u_j - r).$$

If a feasible solution z which satisfies (10) can be found, then a violated GCS inequality will be generated by Proposition 2.

It is easy to see that the relation (10) is equivalent to

$$\sum_{i\in K_1} a_i(u_i - x^*_i)z_i - \sum_{i\in K_2} a_i x^*_i < a'_j(r + 1 + x^*_j - u_j). \tag{11}$$

To develop an efficient strategy for trying to generate the most effective ones, we formulate the following separation problem, which is a mixed integer nonlinear programming (MINLP) problem.

$$\begin{aligned} \xi = \min_{z,r,a'_j} \quad & \sum_{i\in K_1} a_i(u_i - x^*_i)z_i - \sum_{i\in K_2} a_i x^*_i - a'_j(r + 1 + x^*_j - u_j) \\ \text{s.t.} \quad & a'_j = \sum_{i\in K_1} a_i u_i z_i + a_j u_j - b - r a_j \\ & 0 < a'_j < a_j \\ & 0 \le r \le u_j - l_j - 1, r \in \mathbb{Z} \\ & z_i \in \{0, 1\}, \ i \in K_1. \end{aligned} \tag{12}$$

If the optimal value $\xi < 0$, the GCS inequality corresponding to an optimal solution (z, r, a'_j) separates x^*. Otherwise, there exist no violated GCS inequalities while fixing variable x_j. It is not known to us yet whether the problem (12) can efficiently be solved. However, our purpose is not to find the optimal solution of (12) for all the cases but to solve (12) only when $\xi < 0$. The violated information that $\xi < 0$ is crucial and will help us to design a polynomial time algorithm to solve (12).

4.3 Separation algorithm

To develop an efficient algorithm, we shall make use of the violated information that $\xi < 0$ to derive the relationship between r and the current value x^*_j if the variable x_j is fixed as the GCS variable.

Proposition 9 *Consider variable x_j with $j \in I$ and $a_j > 0$. If $x^*_j \le u_j - r - 1$ or $x^*_j \ge u_j - r$, there are no GCS inequalities on variable x_j that cuts off x^*.*

Proof Suppose that some GCS inequality that $a'^T x \leq b'$ is generated on variable x_j with $a_j > 0$. If $x_j^* \leq u_j - r - 1$, it follows that

$$a'^T x^* \leq \overline{\alpha} - a_j u_j + a'_j x_j^* \leq \overline{\alpha} - a_j u_j + a'_j (u_j - r - 1) = b'.$$

Therefore, no GCS inequalities on variable x_j can be generated to cut off x^*. If $x_j^* \geq u_j - r$, it follows from $a'_j < a_j$ that

$$\begin{aligned} a'^T x^* - b' &= a^T x^* + (a'_j - a_j)x_j^* - b + (a_j - a'_j)(u_j - r) \\ &= a^T x^* - b + (a_j - a'_j)(u_j - r - x_j^*) \leq a^T x^* - b \leq 0. \end{aligned}$$

Thus no violated GCS inequalities exist on variable x_j. □

By Proposition 9, we can assume that $\lfloor x_j^* \rfloor = u_j - r - 1$. In this case, r can be regarded as a constant in problem (12). After the elimination of the variable a'_j, problem (12) can be translated into the following mixed integer linear programming problem:

$$\begin{aligned} \min_z \quad & \sum_{i \in K_1} a_k (u_i - x_i^* - u_i f_j^*) z_i - \sum_{i \in K_2} a_i x_i^* + (b + r a_j - a_j u_j) f_j^* \\ \text{s.t.} \quad & 0 < \sum_{i \in K_1} a_i u_i z_i + a_j u_j - b - r a_j < a_j \\ & z_i \in \{0, 1\}, \; i \in K_1 \\ & r = u_j - 1 - \lfloor x_j^* \rfloor, \; f_j^* = x_j^* - \lfloor x_j^* \rfloor. \end{aligned} \tag{13}$$

The direct solution of problem (13) is still hard to compute. To circumvent the difficulty, we provide two more propositions. Let T be the index set of variables with negative objective coefficients in problem (13), namely,

$$T := \{i \in K_1 : u_i - x_i^* - u_i f_j^* < 0\}. \tag{14}$$

Proposition 10 *For a relaxation solution x^* with $x_j^* \notin \mathbb{Z}$, $a_j > 0$ and $j \in I$, if there exists a violated GCS inequality on x_j given by some subset $C \subseteq K_1$ with*

$$a'_j x_j^* + \sum_{i \in C} a_i x_i^* + \sum_{i \in K_2} a_i x_i^* > b',$$

then there also exists a violated GCS inequality on x_j given by some subset $S \subseteq T$.

Proof Assume that there exists a violated GCS inequality given by some subset $C \subseteq K_1$. From the constraints of problem (13),

$$a'_j = \sum_{i \in C} a_i u_i + a_j u_j - b - r a_j.$$

If $\sum_{i \in C \setminus T} a_i u_i < a'_j$, we have that

$$b + r a_j - a_j u_j < \sum_{i \in C \cap T} a_i u_i \leq \sum_{i \in C} a_i u_i < b + (r + 1) a_j - a_j u_j.$$

Therefore, the set $S := C \cap T$ is corresponding to a feasible point of problem (13) and its objective value does not exceed the one related to C.

In the case that $\sum_{i\in C\setminus T} a_i u_i \geq a'_j$, it follows that

$$\sum_{i\in C\cap T} a_i u_i \leq b + ra_j - a_j u_j.$$

The objective value of problem (13) related to C is

$$\begin{aligned}
&\sum_{i\in C} a_i(u_i - x_i^* - u_i f_j^*) - \sum_{i\in K_2} a_i x_i^* + (b + ra_j - a_j u_j) f_j^* \\
\geq &\sum_{i\in C} a_i(u_i - x_i^* - u_i f_j^*) + \sum_{i\in C\cap T} a_i u_i f_j^* - \sum_{i\in K_2} a_i x_i^* \\
= &\sum_{i\in C\setminus T} a_i(u_i - x_i^* - u_i f_j^*) + \sum_{i\in C\cap T} a_i(u_i - x_i^*) - \sum_{i\in K_2} a_i x_i^* \\
\geq &\, 0,
\end{aligned}$$

which contradicts the violated assumption that $\xi < 0$. □

Proposition 11 *If either*

$$\sum_{i\in T} a_i u_i \leq b + ra_j - a_j u_j \text{ or } \sum_{i\in T} a_i u_i \geq b + (r+1)a_j - a_j u_j,$$

then there exist no violated GCS inequalities on variable x_j.

Proof The case that $\sum_{k\in T} a_i u_i \leq b + ra_j - a_j u_j$ is straightforward due to the proof of Proposition 10. Now consider the case $\sum_{i\in T} a_i u_i \geq b + (r+1)a_j - a_j u_j$. Assume that there exists a subset $S \subseteq T$ which leads to a violated GCS inequality. The objective value of problem (13) for S is

$$\begin{aligned}
&\sum_{i\in S} a_i(u_i - x_i^* - u_i f_j^*) + (b + ra_j - a_j u_j) f_j^* - \sum_{i\in K_2} a_i x_i^* \\
\geq &\sum_{i\in T} a_i(u_i - x_i^* - u_i f_j^*) + (b + ra_j - a_j u_j) f_j^* - \sum_{i\in K_2} a_i x_i^* \\
= &\sum_{i\in T} a_i(u_i - u_i f_j^*) + \sum_{i\in T} a_i(-x_i^*) + (b + ra_j - a_j u_j) f_j^* - \sum_{i\in K_2} a_i x_i^* \\
\geq &[b + (r+1)a_j - a_j u_j](1 - f_j^*) - b + a_j x_j^* + (b + ra_j - a_j u_j) f_j^* \\
= &\, 0,
\end{aligned}$$

which contradicts the violated assumption that $\xi < 0$. □

Propositions 10 and 11 indicate that, if having fixed the GCS variable x_j with $a_j > 0$, only the set T [as defined in (14)] is required to be considered in generating the GCS inequalities. More exactly, if the following conditions hold,

$$b + ra_j - a_j u_j < \sum_{i\in T} a_i u_i < b + (r+1)a_j - a_j u_j \tag{15}$$

and

$$\sum_{i\in T} a_i(u_i - x_i^* - u_i f_j^*) - \sum_{i\in K_2} a_i x_i^* + (b + ra_j - a_j u_j) f_j^* < 0, \tag{16}$$

then the most violated inequality based on variable x_j given by T can be generated. Otherwise, we can claim that no GCS inequalities exist on variable x_j.

Now we are able to provide Algorithm 1, which can generate all the most violated GCS inequalities on integer variables with positive coefficients. A similar separation algorithm can be obtained for the case $a_j < 0$, $j \in I$ by Proposition 3.

```
Input: Relaxation solution $x^*$, $a^T x \leq b, l \leq x \leq u, x_j \in \mathbb{Z}, j \in I$
Output: Valid inequalities $\mathcal{F}$ which cut off $x^*$
Normalize the set $M_I$ to satisfy (8); Initialize $\mathcal{F} := \emptyset$;
for all variables $x_j$, $j \in I$ and $a_j > 0$ with $x_j^* \notin \mathbb{Z}$ do
    if (15) and (16) hold then
        Generate the inequality $a'_j x_j + \sum_{i \in T} a_i x_i + \sum_{a_i < 0} a_i x_i \leq b'$;
        Recover the inequality to the original space if necessary;
        Add the violated inequality to $\mathcal{F}$;
    end if
end for
```

Algorithm 1: Separation algorithm for $a^T x \leq b$ on x^*

Proposition 12 *The separation problem* (13) *can be solved by Algorithm 1 in $\mathcal{O}(n|I|)$ operations, where n is the number of nonzero coefficients.*

Proof Notice that there are at most $|I|$ integer variables with $x_j^* \notin \mathbb{Z}$. The work of Algorithm 1 in recovering the inequality to the original space and calculating (15) and (16) can be done in $\mathcal{O}(n)$ operations. Therefore, the separation problem (13) can be solved in $\mathcal{O}(n|I|)$ operations. □

To end this section, we summarize two advantages of GCS inequalities. On one hand, the separation problem can efficiently be solved since the number of nonzero coefficients in one constraint is usually small in real problems. On the other hand, as the GCS inequality is generated by the original constraint, it probably does not contain too large or too small coefficient values. Therefore, these GCS inequalities are usually sparse and stable.

5 Numerical results

We evaluated the performance of GCS inequalities by implementing it as an additional separator in the open source MIP solver SCIP 3.2.1 [22] with the default LP solver SOPLEX 2.2.1 [26]. For convenience, the corresponding separator is called as the GCS separator. All numerical tests described in this section were implemented on a cluster of Intel Core i7-4790 3.60 GHz computer, with 8 MB cache and 8 GB RAM. The operating system is Ubuntu 16.04. According to [27], in order to have a fair comparison, we used the most infeasible branching strategy and disable restart presolving technique and all primal heuristics.

Instances from MIPLIB 3.0 [23], MIPLIB 2003 [24] and the benchmark set of MIPLIB 2010 [25] (denoted by MMM) were used for our numerical experiments. We excluded 3 infeasible instances named *ash608gpia-3col*, *enlight14* and *ns1766074*, and one instance which runs out of memory at the root node in SCIP, namely *mspp16*. This leaves us 164 instances in the MMM test set.

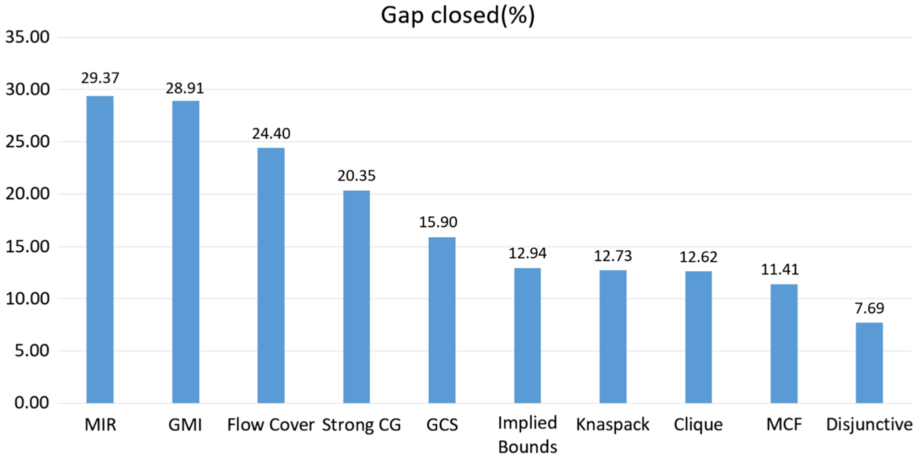

Fig. 1 Gap closed by individual separators

In order to study the performance of the GCS separator more exhaustively, we try to ask the three questions as follows.

(1) What effect does the GCS separator make individually?
(2) Does the GCS separator improve the dual bound when invoking it along with other separators in a MIP solver?
(3) What is the impact of the GCS separator on the overall running time?

If one expects the GCS separator to have some positive effects, the GCS separator must generate some efficient cuts for real application problems. On the other hand, since it can tighten the relaxation set, we want to know how tight it is after the addition of the GCS inequalities. That is the reason why we ask the first question. Since there are some other separators in MIP solvers, we want to know whether the GCS separator is covered by other separators or not. That is the motivation to ask the second question. Finally, by asking the third question, we want to know whether the separator is effective for solving MIP instances.

For the first question, we tested the individual performance of the GCS separator and compared it with some other separators. Since the separators affect one another, we disable all the other separators except the tested one. In order to describe the performance, we use the root gap closed [27] defined as follows:

$$100 \cdot \frac{\check{z} - z_{LP}}{z_{MIP} - z_{LP}},$$

where $\check{z}$ is the dual bound at the root node before branching, z_{LP} is the value of LP relaxation after presolving and z_{MIP} is the optimal (or best known) value of MIP.

Figure 1 shows the arithmetic mean of the gap closed by all separators in SCIP. For the MMM test set, the gap closed by the GCS separator is 15.90%, which is better than Implied Bounds, Knapsack Cover, Clique, MCF and Disjunctive separators but worse than MIR, GMI, Strong CG and Flow Cover separators. This indicates that the GCS separator has a modest effect in improving the dual bound on real data problems. Furthermore, among 164 instances in the MMM test set, it affects 105 instances, more than half of the instances.

For the second question, we tested the performance when invoking it with other separators. As before, the root gap closed is used to evaluate the performance. At this time, we did not

Table 1 Overall gap closed

Setting	All	Affected	Better
	Gap Imp (%)	Gap Imp (%)	
GCS	46.11	58.84	39
No GCS	45.81	58.37	22

Table 2 Time and nodes performances

Setting	Nodes	Time	Solved	Better
GCS	3403.73	41.75	40	8
No GCS	4061.99	51.21	40	6

disable other separators but compared the default setting as described above with the version turning on the GCS separator.

As Table 1 shows, the average root gap closed increases by 0.30% for all 164 instances if the GCS separator is called. For the affected 105 models, the gap closed is improved by 0.47%. Looking at the number of better gap closed instances, it is 39 while the worse cases is only 22. This shows that although the GCS is likely to be covered by other separators in most cases, there exist some cuts which are not generated by the other separators. This result will further be verified at the end of this section.

For the third question, we studied the time and nodes performances after calling the GCS separator in an MIP solver. We only considered those affected instances that can be solved in 3600 s by at least one of the two settings. This leaves us 40 instances.

Table 2 presents the time and nodes performances when adding the GCS separator comparing with the default setting as described above (One can also see Table 3 in "Appendix" for the details). In either case, all of the 40 instances are solved. Interestingly, we see from Table 2 that the geometric mean[1] of time and nodes is significantly improved. Furthermore, 8 instances are at least 10% time faster if calling the GCS separator, while only 6 instances are at least 10% time faster when turning the GCS separator off. This indicates that GCS separator has a positive effect for solving MIP.

To end this section, we illustrate one instance which is much improved by calling GCS separator. The name of such an instance is *quet1*, which was once studied in [28]. We found that the solution time decreases from 342.3 to 3.6 s by calling the GCS separator. Looking at its structure further, it contains the following type of constraints:

$$\sum_{i \in C} a_i x_i \leq y_1 + q y_2,$$

where $x_i \in \mathbb{B}$, $y_1, y_2 \in \mathbb{Z}^+$, $a_i > 0$, $i \in C$ and $q > 1$. Moreover, $1 < \frac{c_{y_2}}{c_{y_1}} < q$, where c_{y_1} and c_{y_1} are the objective coefficients of y_1 and y_2, respectively. Combining all these conditions, the solution of LP relaxation (x^*, y^*) is usually such that $y_2^* \notin \mathbb{Z}$. The GCS separator generates the following type of valid inequalities,

$$\sum_{i \in S} a_i x_i \leq y_1 + q' y_2 + \gamma,$$

where $S \subseteq C$, $0 < q' < q$ and γ is a constant, to separate (x^*, y^*). To the best of our knowledge, these inequalities are hardly generated by the other separators in SCIP. By the

[1] Shifted geometric mean, 10 s for average time and 100 for average nodes [19].

addition of the GCS separator, the gap closed increases from 88.48% to 95.78% and hence leads to a significant improvement in the solution time.

6 Conclusion and future work

We have proposed the family of GCS inequalities by generalizing the idea of the coefficient strengthening method. By studying the invariant property of GCS inequalities, analyzing the variables bounds and using the violated information, we have provided a separation algorithm to find the violated inequalities, which leads to the GCS separator. Our computational results demonstrate that, although the generated GCS cuts are likely to be covered by those generated by some other separators in most cases, there also exist some GCS cuts that are not covered and the proposed GCS separator improves the performance for these instances on average.

There still exist some ideas to be explored to improve the efficiency of GCS cuts. For example, it appears important to aggregate some constraints and use the GCS separator on the aggregated constraint to generate cuts (see [2] for this idea). Comparing with dropping terms directly, the advantage of this technique is that it affects the difference between the current activity and the right hand side as small as possible. Another interesting topic is only to generate the GCS cuts which cannot be generated by other separators, especially by the MIR separator. In our numerical experiments, we observed that the MIR separator generates some identical inequalities as GCS separator generates for some instances including *bell5* and *bab5*. It remains to be explored how to avoid these redundant works.

Appendix

See Table 3.

Table 3 Detail results of nodes and time performance

Instance	GCSC			No GCSC		
	Closed gap (%)	Nodes	Time	Closed gap (%)	Nodes	Time
aflow30a	67.04	51285	82.9	67.97	36,996	**68.1**
bell3a	47.97	32262	5.8	47.97	33,197	5.9
bell5	27.55	4137	0.5	26.46	2148	0.2
binkar10_1	66.54	1,069,939	**940.8**	65.81	1,636,080	1306
bley_xl1	100.00	52	229.2	100.00	12	**172.2**
bnatt350	100.00	41,451	692.3	100.00	48,732	733
dcmulti	81.42	1352	0.7	81.42	1001	0.5
dsbmip	100.00	14	0.2	100.00	63	0.4
fiber	93.18	472	1	95.42	388	0.9
fixnet6	74.34	34,845	22	75.80	29,506	21.4
gen	100.00	1	0	100.00	1	0
gesa2	99.89	33	0.4	99.20	143	0.4
gesa2_o	99.03	83	0.5	98.69	109	0.5

Table 3 continued

Instance	GCSC			No GCSC		
	Closed gap (%)	Nodes	Time	Closed gap (%)	Nodes	Time
gesa3	86.69	121	0.6	86.96	90	0.5
gesa3_o	90.95	79	0.5	89.62	95	0.4
gt2	100.00	4107	0.6	100.00	40	0
khb05250	99.91	11	0.1	99.91	11	0.1
lseu	68.35	821	0.1	60.34	767	0.1
map20	18.91	31,402	3372.6	18.62	31,992	3493.6
mitre	100.00	1	3.6	100.00	1	3.6
mod011	76.13	1274	62.4	72.47	1643	**53.2**
neos-1109824	80.20	529,551	**1158.1**	75.25	1,036,293	2417.3
neos-686190	4.77	335,231	**1074.2**	4.77	472,344	1530.7
noswot	78.46	21,789,353	2902.6	78.46	12,076,736	**1570.6**
ns1208400	0.00	9022	**1234.7**	0.00	9617	1637.7
ns1688347	79.23	32,356	**631.4**	79.27	70,235	1237.6
opt1217	100.00	146	0.5	100.00	44	0.4
p0033	100.00	1	0	100.00	1	0
p0282	76.54	2120	0.7	83.94	2001	0.7
p0548	100.00	1	0.1	100.00	1	0.1
p2756	98.47	1211	1.5	98.51	819	1.1
pg5_34	98.81	606,644	1835.2	98.81	530,422	**1250.8**
pk1	0.00	828,160	122.3	0.00	808,824	120.8
qnet1	95.78	2386	**3.6**	88.48	423,198	342.3
qnet1_o	94.90	632	**1.5**	90.74	244,136	213.8
rentacar	46.83	30	1	46.83	30	1.1
rgn	100.00	1	0.1	100.00	1	0.1
satellites1-25	0.00	6454	759.7	13.33	2072	**464.7**
sp98ir	2.49	158,616	**643.3**	2.49	193049	853.8
vpm2	72.71	1907	0.6	72.59	2166	0.6
all	73.17	3403.73	41.75	73.00	4061.99	51.21

Bold represents the difference in time by more than 10%

References

1. Gomory, R.: An Algorithm for the Mixed Integer Problem. RM-2597. The Rand Corporation, Santa Monica (1960)
2. Marchand, H., Wolsey, L.A.: Aggregation and mixed integer rounding to solve MIPs. Oper. Res. **49**, 363–371 (2001)
3. Letchford, A.N., Lodi, A.: Strengthening Chvátal–Gomory cuts and Gomory fractional cuts. Oper. Res. Lett. **30**, 74–82 (2002)
4. Koster, A.M.C.A., Zymolka, A., Kutschka, M.: Algorithms to separate $\{0, \frac{1}{2}\}$-Chvátal–Gomory cuts. Algorithmica **55**, 375–391 (2008)
5. Hoffman, K.L., Padberg, M.: Improving LP-representations of zero-one linear programs for branch-and-cut. ORSA J. Comput. **3**, 121–134 (1991)
6. Crowder, H., Johnson, E.L., Padberg, M.: Solving large-scale zero-one linear programming problems. Oper. Res. **31**, 803–834 (1983)

7. Atamtürk, A., Nemhauser, G.L., Savelsbergh, M.W.P.: Conflict graphs in solving integer programming problems. Eur. J. Oper. Res. **121**, 40–55 (2000)
8. Balas, E.: Disjunctive programming. Ann. Discret. Math. **5**, 3–51 (1979)
9. Weismantel, R.: On the 0/1 knapsack polytope. Math. Program. **77**, 49–68 (1997)
10. Van Roy, T.J., Wolsey, L.A.: Valid inequalities and separation for uncapacitated fixed charge networks. Oper. Res. Lett. **4**, 105–112 (1985)
11. Padberg, M.W., Van Roy, T.J., Wolsey, L.A.: Valid linear inequalities for fixed charge problems. Oper. Res. **33**, 842–861 (1985)
12. Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P.: Lifted flow cover inequalities for mixed 0–1 integer programs. Math. Program. **85**, 439–467 (1999)
13. Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P.: Lifted cover inequalities for 0–1 integer programs: computation. INFORMS J. Comput. **10**, 427–437 (1998)
14. Achterberg, T., Raack, C.: The Mcf-separator: detecting and exploiting multi-commodity flow structures in MIPs. Math. Program. Comput. **2**, 125–165 (2010)
15. Achterberg, T., Wunderling, R.: Mixed integer programming: analyzing 12 years of progress. In: Jnger, M., Reinelt, G. (eds.) Facets of Combinatorial Optimization, pp. 449–481. Springer, Berlin (2013)
16. IBM CPLEX Optimizer. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html
17. Achterberg, T., Bixby, R.E., Gu, Z., Rothberg, E., Weninger, D.: Presolve reductions in mixed integer programming. ZIB Report 16–44, Zuse Institute Berlin, (2016)
18. Savelsbergh, M.W.P.: Preprocessing and probing techniques for mixed integer programming problems. ORSA J. Comput. **6**, 445–454 (1994)
19. Achterberg, T.: Constraint Integer Programming. Ph.D. thesis, Technische Universität, Berlin (2007). https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/1112
20. Atamtürk, A., Rajan, D.: On splittable and unsplittable flow capacitated network design arc-set polyhedra. Math. Program. **92**, 315–333 (2002)
21. Magnanti, T.L., Mirchandani, P., Vachani, R.: The convex hull of two core capacitated network design problems. Math. Program. **60**, 233–250 (1993)
22. Gamrath, G., Fischer, T., Gally, T., Gleixner, A.M., Hendel, G., Koch, T., Maher, S.J., Miltenberger, M., Müller, B., Pfetsch, M.E., Puchert, C.: The SCIP optimization suite 3.2. ZIB Report 15–60, Zuse Institute Berlin, (2016)
23. Bixby, R., Ceria, S., McZeal, C., Savelsbergh, M.: An updated mixed integer programming library: MIPLIB 3.0. Optima **54**, 12–15 (1998)
24. Achterberg, T., Koch, T., Martin, A.: MIPLIB 2003. Oper. Res. Lett. **34**, 361–372 (2006)
25. Koch, T., Achterberg, T., Andersen, E., Bastert, O., Berthold, T., Bixby, R.E., Danna, E., Gamrath, G., Gleixner, A.M., Heinz, S., Lodi, A., Mittelmann, H., Ralphs, T., Salvagnin, D., Steffy, D.E., Wolter, K.: MIPLIB 2010. Math. Program. Comput. **3**, 103–163 (2011)
26. Wunderling, R.: Paralleler und objektorientierter simplex. Ph.D. thesis, Technische Universität Berlin, (1996). https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/538
27. Wolter, K.: Implementation of cutting plane separators for mixed integer programs. Master's thesis, Technische Universität Berlin, (2006)
28. Beate, B., Günlük, O., Wolsey, L.A.: Designing private line networks: polyhedral analysis and computation. CORE Discussion Paper 9647, Université Catholique de Louvain, (1996)