Numerically Accurate Gomory Mixed-Integer Cuts

William Cook*
Industrial and Systems Engineering
Georgia Institute of Technology

Sanjeeb Dash
Discrete Optimization Group
IBM T. J. Watson Research Center

Ricardo Fukasawa
Industrial and Systems Engineering
Georgia Institute of Technology

Marcos Goycoolea†
School of Business
Universidad Adolfo Ibáñez, Chile

March 4, 2008

Abstract

We describe a simple process for generating numerically accurate cutting planes using floating-point arithmetic and the mixed-integer rounding (MIR) procedure. Applying this method to the rows of the simplex tableau permits the generation of Gomory mixed-integer cuts that are guaranteed to be satisfied by all feasible solutions to a mixed-integer programming problem. We report on tests with the MIPLIB 3.0 and MIPLIB 2003 test collections, and with MIP instances derived from the TSPLIB traveling salesman library.

1 Introduction

Mixed-integer programming (MIP) is a fundamental tool in operations research. A MIP problem has the form

$$\begin{gathered} \text{Minimize } c^T x \\ \text{subject to} \\ Ax \geq d,\ l \leq x \leq u,\ x_j \text{ integer } (j \in I) \end{gathered} \tag{1}$$

where A is an $m \times n$ matrix, d is an m-vector, c, l, u are n-vectors, x is an n-vector of variables, and $I \subseteq \{1, \ldots, n\}$. Introductions to MIP theory and applications can be found in Nemhauser and Wolsey (1988), Wolsey (1998), and Bertsimas and Weismantel (2005).

The past decade has seen rapid improvements in the quality of software for the solution of general MIP problems. Bixby et al. (2000, 2004) report that a critical factor in this progress has been the increased use of cutting planes, or *cuts*, to improve the linear programming (LP) relaxations of MIP models. An important milestone in this context is the study of Balas et al. (1996), renewing interest in the computational application of Gomory mixed-integer cuts (Gomory 1960). Indeed, among the varieties of cutting planes in current use,

*Supported by NSF Grant CMMI-0726370 and ONR Grant N00014-03-1-0040.
†Supported by FONDECYT Grant 11075028.

Bixby et al. (2004) rank Gomory cuts as the class having the greatest overall impact on the performance of MIP solvers.

Gomory cuts are obtained via a process known as mixed-integer rounding (MIR), applied to the rows of an optimal simplex tableau (Wolsey 1998). It is important to note that both the MIR process and the linear algebra required to construct a simplex tableau row are subject to rounding errors when computed in floating-point arithmetic, the standard platform adopted in MIP software. This numerical difficulty can, in practice, lead to the generation of cutting planes that are not valid for a given MIP instance. Indeed, in a careful study, Margot (2007) documents the frequent occurrence of infeasible cuts in applications of the MIR procedure. A consequence of this numerical difficulty is that MIP algorithms can become unstable if Gomory cuts are added in multiple rounds, where cuts become a part of the LP relaxation used to produce further cuts. For this reason, commercial MIP solvers limit the repeated application of Gomory cuts, even in cases where significant progress in the objective bound of the LP relaxation is being made. This is the topic we address in our current study.

A cutting-plane generation process is called *safe* in a specified model of arithmetic if it is guaranteed to produce only inequalities that are valid for all feasible solutions to a given MIP problem. Neumaier and Shcherbina (2004) introduce the notion of safe cuts and use a combination of interval arithmetic and directed rounding to derive a safe version of the MIR procedure for floating-point computation. In this paper we present an alternative implementation of safe MIR cuts in floating-point arithmetic and we demonstrate its application in generating Gomory cuts in general MIP instances. We report on computational studies with the MIPLIB 3.0 and MIPLIB 2003 test collections (Bixby et al. 1998 and Achterberg et al. 2006) and with MIP instances derived from the TSPLIB traveling salesman library (Reinelt 1991). In the TSPLIB tests, our method uses the Concorde TSP solver (Applegate et al. 2006) and safe Gomory cuts to establish lower bounds on the lengths of tours.

Our computer code is based on standard IEEE floating-point arithmetic (IEEE 1985), and the methods can be easily incorporated into existing MIP solvers. The source code is available at

`www.isye.gatech.edu/~rfukasaw/safemir`

as part of a package for experimental work in MIP cutting-plane methods.

The paper is organized as follows. In Section 2 we define the MIR procedure for generating a cut from a mixed-integer knapsack set. General properties of floating-point arithmetic are discussed in Section 3, together with simple procedures for safe row aggregation and safe substitution of slack variables. In Section 4 we present the safe MIR procedure and in Section 5 we extend this to a safe procedure for the more general complemented-MIR cuts. The Neumaier-Shcherbina method for obtaining valid dual bounds for bounded LP problems is discussed in Section 6. Finally, in Section 7 we present the results of our computational study.

2 MIR inequalities

The sets of rational numbers, real numbers, and integers are denoted by $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{Z}$, respectively. For $t \in \mathbb{R}$, $\lfloor t \rfloor$ denotes the greatest integer less than or equal to t, $\lceil t \rceil$ denotes the smallest integer greater than or equal to t, and $\hat{t}$ denotes $t - \lfloor t \rfloor$.

Let n be a positive integer and let N denote the set $\{1, \ldots, n\}$. Consider $a \in \mathbb{Q}^n$, $b \in \mathbb{Q}$, and a partition (I, C) of N. A single-row relaxation of an MIP model can take the form

$$K = \{x \in \mathbb{R}^n : \sum_{j=1}^{n} a_j x_j \geq b,\ x \geq 0,\ x_j \in \mathbb{Z}\ \forall j \in I\}. \tag{2}$$

The defining inequality of K can, for example, be one of the original MIP constraints or, more generally, a nonnegative linear combination of the original constraints.

Let $S = \{j \in I : \hat{a_j} \leq \hat{b}\}$. The *MIR inequality*

$$\sum_{j \in S} (\hat{a_j} + \hat{b}\lfloor a_j \rfloor) x_j + \sum_{j \in I \setminus S} (\hat{b} + \hat{b}\lfloor a_j \rfloor) x_j + \sum_{j \in C} \max\{a_j, 0\} x_j \geq \hat{b}\lceil b \rceil \tag{3}$$

is valid for K and can therefore be considered for use as a cutting plane to improve the LP relaxation of the original MIP model (Wolsey 1998).

Defining $f(q_1, q_2) := \min\{\hat{q_1}, \hat{q_2}\} + \hat{q_2}\lfloor q_1 \rfloor$ and $h(q) := \max\{q, 0\}$, the MIR inequality (3) can be written as

$$\sum_{i \in I} f(a_i, b) x_i + \sum_{i \in C} h(a_i) x_i \geq \hat{b}\lceil b \rceil. \tag{4}$$

This compact notation will be convenient in our discussion of accurate versions of the inequality.

For each $j \in C$, let $l_j \in \mathbb{Q}$ and $u_j \in \mathbb{Q} \cup \{+\infty\}$. For each $j \in I$, let $l_j \in \mathbb{Z}$ and $u_j \in \mathbb{Z} \cup \{+\infty\}$. Assume $0 \leq l_j \leq u_j$ for each $j \in N$ and consider the set

$$K_B = \{x \in \mathbb{R}^n : \sum_{i \in N} a_j x_j \geq b,\ l_j \leq x_j \leq u_j\ \forall j \in N \text{ and } x_j \in \mathbb{Z}\ \forall j \in I\}. \tag{5}$$

The lower and upper bounds on the variables lead to a slightly more general form of the MIR inequality, called the *complemented-MIR* (c-MIR) inequalities (Marchand and Wolsey 2001). The name derives from the fact that the c-MIR inequalities are obtained by complementing variables.

Let U, L be disjoint subsets of N such that every $j \in U$ satisfies $u_j < \infty$. By substituting variables x_j with $j \in U$ by $u_j - x_j$, and variables x_j with $j \in L$ by $x_j - l_j$, applying the MIR procedure, and substituting back, we obtain the following c-MIR inequality for K_B:

$$-\sum_{U \cap I} f(-a_j, b) x_j + \sum_{I \setminus U} f(a_j, r) x_j - \sum_{U \cap C} h(-a_j) x_j + \sum_{C \setminus U} h(a_j) x_j \geq R, \tag{6}$$

where

$$R = \hat{b}\lceil b \rceil - \sum_{U \cap I} f(-a_j, b) u_j + \sum_{L \cap I} f(a_j, b) l_j - \sum_{U \cap C} h(-a_j) u_j + \sum_{L \cap C} h(a_j) l_j.$$

Marchand and Wolsey (2001) have demonstrated the effectiveness of c-MIR cuts in practical computations with MIP test instances. Further computational results on MIR and c-MIR cuts are reported in Balas and Saxena (2006), Dash et al. (2006), Goycoolea (2006), and Dash et al. (2007).

If the defining inequality of K_B is taken as a row of an optimal simplex tableau for the LP relaxation of an MIP instance, then the c-MIR cut is a form of the Gomory mixed-integer cut. Our computational study will focus on Gomory cuts, but the numerically safe methods will be presented in the general context of the MIR and c-MIR procedures.

3 Floating-point arithmetic

The validity of MIR and c-MIR inequalities, for the sets K and K_B, relies on the correctness of arithmetic calculations, which cannot be guaranteed when floating-point arithmetic is employed. Moreover, if the defining inequality for K or K_B is obtained by aggregating several of the original constraints from an MIP instance, then caution must be taken to ensure that the set itself is a valid relaxation. To discuss an approach for dealing with these issues, we give a brief description of the floating-point-arithmetic platform.

3.1 The model of floating point-arithmetic

A floating-point number consists of a sign, exponent, and mantissa. Standard IEEE double precision floating-point arithmetic allots 11 bits for the exponent and 52 bits for the mantissa (Goldberg 1991). In our discussion we assume only that the number of bits assigned to the exponent and mantissa are fixed to some known values.

With the above assumption, let $\mathbb{M} \subseteq \mathbb{Q}$ be the set of floating point-representable reals. Note that $\mathbb{M}$ is finite. If $q \in \mathbb{M}$, then both $-q$ and $\hat{q}$ are members of M. Also, if $a, b \in \mathbb{M}$, then $\min\{a, b\}$ and $\max\{a, b\}$ are members of M.

The set $\mathbb{M}$ does not have much structure; it is not a monoid since associativity for addition does not hold.

3.2 Approximating real functions

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, we say that a function $f^{up} : \mathbb{M}^n \to \mathbb{M}$ upper approximates f in $\mathbb{M}$ if

$$f^{up}(x) \geq f(x) \text{ if } x \in \mathbb{M}^n$$

and we say that a function $f^{dn} : \mathbb{M}^n \to \mathbb{M}$ lower approximates f in $\mathbb{M}$ if

$$f^{dn}(x) \leq f(x) \text{ if } x \in \mathbb{M}^n.$$

Note that the basic arithmetic operations $+$ and $*$ are functions from $\mathbb{R}^2 \to \mathbb{R}$. Let $+^{up} : \mathbb{M}^2 \to \mathbb{M}$ and $+^{dn} : \mathbb{M}^2 \to \mathbb{M}$ be upper and lower approximations of $+$, and let $*^{up} : \mathbb{M}^2 \to \mathbb{M}$ and $*^{dn} : \mathbb{M}^2 \to \mathbb{M}$ be upper and lower approximations of $*$. (In the C programming language (ISO/IEC 1999), IEEE floating-point rounding conventions can be set to produce upper and lower approximations using the `fesetround` function. Upper approximations are obtained by first calling the function with the argument `FE_UPWARD` and then carrying out the usual arithmetic operation. Similarly, lower approximations are obtained using the `FE_DOWNWARD` argument.)

To simplify notation, we write

$$\overline{a+b} := +^{up}(a, b) \text{ and } \underline{a+b} := +^{dn}(a, b)$$

for addition and similarly for subtraction and multiplication.

Let $n \geq 3$ be an integer and let $a_i \in \mathbb{M}$, $\pi_i \in \mathbb{M}$ for $i = 1, \ldots, n$. We define upper and lower approximations for $\sum_{i=1}^{n} \pi_i a_i$ by

$$\overline{\sum_{i=1}^{n} \pi_i a_i} := \overline{\left(\overline{\sum_{i=1}^{n-1} \pi_i a_i} \right) + \overline{\pi_n a_n}}$$

and

$$\underline{\sum_{i=1}^{n} \pi_i a_i} := \underline{\left(\underline{\sum_{i=1}^{n-1} \pi_i a_i} \right) + \underline{\pi_n a_n}}$$

respectively. It is important to notice that any ordering of $a_1, \ldots, a_n$ will yield an upper/lower approximation of $\sum_{i=1}^{n} \pi_i a_i$, but different orders may yield different results. Thus, the operation is not commutative.

3.3 Safe row aggregation

Consider a polyhedron

$$P = \{x \in \mathbb{R}^n : Ax \geq d, l \leq x \leq u\}$$

with $A \in \mathbb{M}^{m \times n}$, $d \in \mathbb{M}^m$, $u \in \{\mathbb{M} \cup \{+\infty\}\}^n$, $l \in \mathbb{M}^n$, and $0 \leq l \leq u$. Given a set of multipliers $\lambda \in \mathbb{M}^m$, with $\lambda \geq 0$, the inequality $\lambda^T Ax \geq \lambda^T d$ is satisfied by all $x \in P$, but this may not be true in the presence of rounding errors. Nonetheless, since $x \geq 0$, we have that

$$\sum_{j=1}^{n} \overline{\left(\sum_{i=1}^{m} \lambda_i a_{ij} \right)} x_j \geq \underline{\sum_{i=1}^{m} \lambda_i d_i}$$

is always a valid inequality. This process permits the safe aggregation of systems of inequalities.

The same concept can be applied to equality systems $Ax = d$, but in this case the multipliers λ need not be restricted to nonnegative values.

3.4 Safe substitution of slack variables

Suppose $P \subseteq \mathbb{R}^{n+1}$ is a polyhedron such that every point $(x, s) \in P$ satisfies the equation

$$\sum_{j=1}^{n} c_i x_i + \beta s = d \tag{7}$$

with $\beta \in \{-1, +1\}$. Consider an inequality

$$\sum_{j=1}^{n} \pi_j x_j + \rho s \geq \pi_o \tag{8}$$

that is valid for P. We wish to eliminate s from (8), while maintaining its validity.

Note that adding $-\rho\beta$ times (7) to (8) gives the inequality

$$\sum_{j=1}^{n}(\pi_j - \rho\beta c_j)x_j + (\rho - \rho)s \geq \pi_o - \rho\beta d.$$

Assuming all coefficients in (7) and (8) are in $\mathbb{M}$, this suggests that we apply safe row aggregation using $\lambda = -\rho\beta \in \mathbb{M}$ for (7) and $\lambda = 1 \in \mathbb{M}$ for (8). The coefficient of the s variable in the resulting inequality is

$$\overline{\overline{1\rho} + \overline{(-\rho\beta)\beta}} = \overline{\rho + \overline{-\rho\beta^2}} = \overline{\rho + \overline{-\rho}} = \overline{\rho - \rho} = 0$$

and thus we have safely removed s from (8).

4 Safe MIR inequalities

For the single-row relaxation (2) in Section 2, we described the MIR inequality

$$\sum_{i\in I} f(a_i, b)x_i + \sum_{i\in C} h(a_i)x_i \geq \hat{b}\lceil b\rceil$$

where $f(q_1, q_2) := \min\{\hat{q_1}, \hat{q_2}\} + \hat{q_2}\lfloor q_1\rfloor$ and $h(q) := \max\{q, 0\}$. To obtain a safe version of this cutting plane in floating-point arithmetic, define the upper approximation f^{up} : $\mathbb{M} \times \mathbb{M} \to \mathbb{M}$ of f as

$$f^{up}(q_1, q_2) := \overline{\left(\min\{\hat{q_1}, \hat{q_2}\} + \overline{\hat{q_2}\lfloor q_1\rfloor}\right)}$$

and define $h^{up}(q) := h(q)$ (no rounding is needed).

Since $x \geq 0$, we have

$$\sum_{i\in I} f^{up}(a_i, b)x_i + \sum_{i\in C} h^{up}(a_i)x_i \geq \sum_{i\in I} f(a_i, b)x_i + \sum_{i\in C} h(a_i)x_i \geq \hat{b}\lceil b\rceil \geq \underline{\hat{b}\lceil b\rceil}.$$

Therefore

$$\sum_{i\in I} f^{up}(a_i, b)x_i + \sum_{i\in C} h^{up}(a_i)x_i \geq \underline{\hat{b}\lceil b\rceil} \tag{9}$$

is a safe MIR inequality.

5 Safe c-MIR inequalities

Consider again the set

$$K_B = \{x \in \mathbb{R}^n : \sum_{i\in N} a_j x_j \geq b,\ l_j \leq x_j \leq u_j\ \forall j \in N \text{ and } x_j \in \mathbb{Z}\ \forall j \in I\}$$

defined in Section 2.

Let sets U, S, L form a partition of $N = \{1, \ldots, n\}$, such that every $j \in U$ satisfies $u_j < \infty$ and every $j \in S$ satisfies $l_j \geq 0$. For each $j \in U$, define $x_j^u = u_j - x_j$. Also, define

$x^l_j = x_j - l_j$ for each $j \in L$ and $x^l_j = x_j$ for each $j \in S$. Observe that x^u_j and x^l_j are both bounded below by zero. Furthermore, if $j \in I$ and x_j takes an integer value, so will x^u_j or x^l_j, depending on if $j \in U$ or $j \in L \cup S$.

By substituting variables x_j with $j \in U$ by $u_j - x^u_j$, variables x_j with $j \in L$ by $x^l_j + l_j$ and denoting $x^l_j = x_j$ for $j \in S$, we obtain the inequality

$$\sum_{j\in U}(-a_j)x^u_j + \sum_{j\notin U} a_j x^l_j \geq b - \sum_{j\in U} a_j u_j - \sum_{j \in L} a_j l_j. \tag{10}$$

Now note that regardless of the sign of a_j, u_j and l_j, we have $\overline{a_j u_j} \geq a_j u_j$ and $\overline{a_j l_j} \geq a_j l_j$. It follows that

$$\overline{\sum_{j\in U} a_j u_j} \geq \sum_{j\in U} \overline{a_j u_j} \geq \sum_{j\in U} a_j u_j$$

and

$$\overline{\sum_{j\in L} a_j l_j} \geq \sum_{j\in L} \overline{a_j l_j} \geq \sum_{j\in L} a_j l_j$$

and thus we have the valid inequality

$$\sum_{j\in U}(-a_j)x^u_j + \sum_{j\notin U} a_j x^l_j \geq \underline{b - \overline{\sum_{j\in U} a_j u_j} - \overline{\sum_{j \in L} a_j l_j}}. \tag{11}$$

Observe that $x^u_j, x^l_j \geq 0$ for all $j \in N$, hence we can apply the MIR procedure to obtain a valid inequality for (11) subject to the corresponding nonnegativity constraints.

Let $r = \underline{b - \overline{\sum_{j\in U} a_j u_j} - \overline{\sum_{j\in L} a_j l_j}} \in \mathbb{M}$. If we apply (9) we obtain

$$\sum_{U\cap I} f^{up}(-a_j, r)x^u_j + \sum_{I\setminus U} f^{up}(a_j, r)x^l_j + \sum_{U\cap C} h^{up}(-a_j)x^u_j + \sum_{C\setminus U} h^{up}(a_j)x^l_j \geq \underline{\hat{r}\lceil r\rceil}.$$

If we substitute back the variables we get the following valid inequality in terms of our original variables:

$$\begin{aligned}\sum_{S\cap I} f^{up}(a_j, r)x_j + \sum_{U\cap I} f^{up}(-a_j, r)(u_j - x_j) + \sum_{L\cap I} f^{up}(a_j, r)(x_j - l_j)& \\ + \sum_{S\cap C} h^{up}(a_j)x_j + \sum_{U\cap C} h^{up}(-a_j)(u_j - x_j) + \sum_{L\cap C} h^{up}(a_j)(x_j - l_j) &\geq \underline{\hat{r}\lceil r\rceil}.\end{aligned}$$

Rearranging the terms we obtain

$$\begin{aligned}&\sum_{U\cap I} f^{up}(-a_j, r) * (-1)x_j + \sum_{I\setminus U} f^{up}(a_j, r)x_j \\ &\quad + \sum_{U\cap C} h^{up}(-a_j) * (-1)x_j + \sum_{C\setminus U} h^{up}(a_j)x_j \geq \underline{\hat{r}\lceil r\rceil} + \\ &\quad \sum_{U\cap I} f^{up}(-a_j, r) * (-1)u_j + \sum_{L\cap I} f^{up}(a_j, r)l_j \\ &\qquad + \sum_{U\cap C} h^{up}(-a_j) * (-1)u_j + \sum_{L\cap C} h^{up}(a_j)l_j.\end{aligned}$$

Now regardless of the signs of u_j, l_j, we have

$$\underline{\hat{r}\lceil r\rceil} + \sum_{U\cap I} f^{up}(-a_j, r) * (-1)u_j + \sum_{L\cap I} f^{up}(a_j, r)l_j + \sum_{U\cap C} h^{up}(-a_j) * (-1)u_j + \sum_{L\cap C} h^{up}(a_j)l_j \geq R$$

where

$$R = \underline{\hat{r}\lceil r\rceil} + \underline{\sum_{U\cap I}(f^{up}(-a_j, r) * (-1))u_j} + \underline{\sum_{L\cap I} f^{up}(a_j, r)l_j} + \underline{\sum_{U\cap C}(h^{up}(-a_j) * (-1))u_j} + \underline{\sum_{L\cap C} h^{up}(a_j)l_j}.$$

Thus

$$\sum_{U\cap I} f^{up}(-a_j, r) * (-1)x_j + \sum_{I\setminus U} f^{up}(a_j, r)x_j + \sum_{U\cap C} h^{up}(-a_j) * (-1)x_j + \sum_{C\setminus U} h^{up}(a_j)x_j \geq \underline{R}$$

is a valid inequality for K_B with all coefficients in $\mathbb{M}$.

6 Safe dual bounds

In this section we describe the technique of Neumaier and Shcherbina (2004) for computing valid dual LP bounds using floating-point arithmetic.

Consider the bounded LP

$$\min\{c^T x : Ax \geq b,\ 0 \leq l \leq x \leq u\} \tag{12}$$

and its dual

$$\max\{b^T y + l^T w - u^T z : A^T y + w - z = c,\ y, w, z \geq 0\} \tag{13}$$

where we assume all components of u are finite.

If (12) is a relaxation of an MIP instance, then an optimal solution provides a bound on the MIP objective value. The problem we face is that because of intrinsic errors in floating-point arithmetic, obtaining an optimal solution to (12) may be difficult. Note, however, that any feasible solution to the dual (13) will also give a bound via the LP weak duality theorem.

Consider any assignment of (potentially infeasible) values y^* to the dual variables y. If $A^T y^* = c$, then $b^T y^*$ is a lower bound for (12). However, ensuring that $A^T y^* = c$ may itself be difficult in floating-point arithmetic. We get around this problem by taking advantage of the form of (13).

For $j \in \{1, \ldots, n\}$, the jth constraint of (13) is $\sum_{i=1}^{m} y_i a_{ij} + w_j - z_j = c_j$. Given y^*, we have $w_j - z_j = c_j - \sum_{i=1}^{m} y_i^* a_{ij}$. Set $z_j^* = \max\{0, \overline{\sum_{i=1}^{m} y_i^* a_{ij} - c_j}\}$. Clearly, $z_j^* \in \mathbb{M}$ and

$$z_j^* \geq \overline{\sum_{i=1}^{m} y_i^* a_{ij} - c_j} \geq \sum_{i=1}^{m} y_i^* a_{ij} - c_j.$$

Thus, we can set $w_j^* = z_j^* - (\sum_{i=1}^{m} y_i^* a_{ij} - c_j) \geq 0$.

It is difficult to guarantee that $w_j^* \in \mathbb{M}$, but since $l \geq 0$ we have

$$\sum_{i=1}^{m} b_i y_i^* + \sum_{j=1}^{n} l_j w_j^* - \sum_{j=1}^{n} u_j z_j^* \geq \sum_{i=1}^{m} b_i y_i^* + \sum_{j=1}^{n} (-u_j) z_j^* \geq \underline{\sum_{i=1}^{m} b_i y_i^* + \sum_{j=1}^{n} (-u_j) z_j^*}$$

To obtain a good lower bound, the values y^* can be taken as the optimal dual solution returned by a floating-point LP solver. Such values y^* are likely to be only slightly infeasible. Note that the bound we obtain via the safe procedure may be less than the floating-point solver's claimed objective value, due to possible increases in z^* and to dropping the w^* variables.

This approach is similar to the bounding technique adopted in the TSP code of Applegate et al. (2006). In their work, the dual values y^* are truncated to a fixed precision, appropriate fixed-precision values are assigned to w and z, and the dual bound is computed in the same fixed precision with overflow checking enabled. An advantage of the method described above is that the bound can be computed entirely in floating-point arithmetic.

7 Computational study

In our study we consider two scenarios where safe MIR procedures can potentially be useful. The first set of tests is concerned with the use of safe cuts in cases that demand accurate bounds. We use the TSP as a case study, applying multiple rounds of safe Gomory cuts to improve LP relaxations generated by TSP-specific methods.

The second tests consider the repeated application of Gomory cuts for general MIP instances. Generating Gomory cuts based on previous cuts can quickly lead to inaccurate results with standard MIP software. Our tests aim to give an indication of the possible improvements in LP bounds that can be obtained by adopting the safe methods, where multiple rounds of cuts can be added without loss of accuracy.

Note that as an alternative to our safe MIR approach, the computations could be carried out entirely in exact rational arithmetic, using the LP solver of Applegate et al. (2007) and rational versions of the MIR procedure. On large instances such an approach is considerably more time-consuming, however, due to the complexity of rational computations on the difficult LP instances that are created. Moreover, the approach adopted here demonstrates the feasibility of including safe MIR methods in standard floating-point-based software.

7.1 Selection of Gomory cuts

Given an MIP instance (1), our cut-generation process begins by adding a slack variable to each row of the model to obtain equality constraints. If a row is such that all of the participating variables are integer, all of the left-hand-side coefficients are integer, and the right-hand-side is integer, then the corresponding slack variable is defined to be integer.

The resulting LP relaxation is solved with the simplex algorithm, producing a basic optimal solution x^*. We improve the relaxation by adding rounds of Gomory cuts, where each round consists of the following steps.

1. *Variable complementation.* Define the index sets $U = \{j : x_j^* = u_j, 1 \leq j \leq n\}$ and $L = \{j : x_j^* = l_j, 1 \leq j \leq n\}$.

2. *Rank the fractional variables.* The integer variables x_j that take on a fractional value x_j^* in the current basis are ranked in non-decreasing value of $|x_j^* - 0.5|$.

3. *Row selection.* Select the 500 highest-ranking variables and safely compute the corresponding tableau rows.

4. *Compute the cuts.* Process the selected rows in the ranked order. For each row safely compute the c-MIR inequality using the sets U and L as defined above, after scaling by $K = 1$, $K = 2$, and $K = 3$, as proposed in Cornuéjols et al. (2003). Stop this procedure if 500 (sufficiently) violated cuts are generated before all selected variables are processed.

5. *Add the cuts.* Add the computed cuts to the LP and resolve.

6. *Remove cuts.* Remove from the LP all previously added cuts that are no longer violated.

This is a straightforward implementation of Gomory cuts. For a discussion of general cut selection see Achterberg (2007) and Goycoolea (2006).

7.2 TSPLIB results

Early cutting-plane research on the TSP by Martin (1966), Miliotis (1978), and others adopted general-purpose MIP codes for improving LP relaxations. In later work, these methods were replaced by algorithms for generating TSP-specific cutting planes. With the safe MIR procedures it is possible to combine these ideas, employing general cuts to further improve relaxations obtained by problem-specific methods.

We consider MIP relaxations obtained by long runs of the Concorde TSP solver (Applegate et al. 2006). In Table 1 we give statistics for the relaxations for all TSPLIB instances having at least 3,000 cities, with the exception of fl3795. In the case of fl3795, the Concorde LP bound establishes the optimality of the tour. Except for pla33810 and pla85900, the set of variables in each case is obtained by reduced-cost fixing based on the Concorde relaxation (the variables set to 0 are removed from the LP), thus any LP bound obtained with general MIP cuts is a valid bound for the original TSP. In the cases of the two largest instances, pla33810 and pla85900, only a subset of the variables is given due to the large number that remain after reduced-cost fixing.

Table 1: MIP Relaxations of the TSP

Name	Variables	TSP Optimal	LP Value	+Cuts	Gap Closed
pcb3038	6976	137694	137684.25	137684.52	2.51%
fnl4461	10129	182566	182558.55	182559.97	19.08%
rl5915	24939	565530	565484.03	565487.91	8.44%
rl5934	25285	556045	555994.47	556001.02	12.97%
pla7397	27209	23260728	23258946.65	23259208.71	14.71%
rl11849	70259	923288	923208.71	923210.01	1.53%
usa13509	129837	19982859	19981199.08	19981229.43	1.83%
brd14051	83221	469385	469353.80	469353.91	0.35%
d15112	110072	1573084	1572966.32	1572967.08	0.64%
d18512	141025	645238	645194.86	645195.17	0.71%
pla33810	67683 (+)	660048945	66001233.03	66001901.19	1.40%
pla85900	167870 (+)	142382641	142296659.63	142299299.63	3.07%

Table 2: Valid MIP Relaxation for pla85900

Name	Variables	TSP Optimal	LP Value	+Cuts	Gap Closed
pla85900f	300969	142382641	142381453.65	142381460.98	0.62%

The relaxations were found with `concorde -mC48 -Z3`, the strongest recommend setting of the Concorde separation routines. This setting uses repeated local cuts, up to size 48, as well as the domino-parity inequalities (see Applegate et al. 2006).

The optimal value of the Concorde LP for each instance is reported in the "LP Value" column of Table 1. In the "+Cuts" column we report the improved lower bounds obtained by multiple rounds of safe Gomory cuts, after applying the safe-LP bounding technique from Section 6. The improvements range from 0.35% of the Concorde LP optimality gap for brd14051, up to 19.09% of the gap for fnl4461. It is interesting, however, that the very strong LP relaxations obtained by Concorde could be improved with general-purpose MIP cuts, suggesting that this may be a technique worth considering for problem classes that have not received the intense study given to the TSP.

To further illustrate the use of of the safe MIR procedure, we constructed a valid MIP relaxation for the 85900-city TSP instance pla85900, using the full set of variables that were not eliminated by reduced-cost fixing. This much stronger relaxation was obtained from the very long computation described in Applegate at al. (2006). The result after adding Gomory cuts to this instance is reported in Table 2, showing a 0.62% increase in the lower bound. Although this is a modest increase, it is possible that the LP solution is sufficiently altered to permit the further use of Concorde's TSP-specific cutting planes in an iterative fashion. (We have not tested this idea.)

7.3 MIPLIB results

To test the effectiveness of repeated rounds of Gomory cuts in general, we constructed a test set consisting of the union of the following instances.

- The full set of 65 instances in the MIPLIB 3.0 collection (Bixby et al. 1998).
- The 27 instances in the MIPLIB 2003 collection (Achterberg et al. 2006) that are not included in MIPLIB 3.0.
- The 13 TSPLIB-MIP instances described in Section 7.2

We excluded four of these instances due to the presence of variables that may assume negative values (that are not handled in our current implementation), leaving a total of 101 instances.

To illustrate the impact of multiple rounds of safe cutting planes, we recorded the lower bounds obtained after 1, 2, 4, 8, 16, and 128 rounds of our cutting-plane procedure; if the optimal LP basis does not change after a round of cutting planes, then the particular run is terminated. In these tests, three of the instances were excluded since they each have zero integrality gap, leaving a total of 98 examples. The results are displayed as curves in Figure 1. The points (x, y) in each curve represent how many instances y closed at least x fraction of the starting integrality gap, after adding the indicated number of rounds of cuts. For example, after one round of cuts 50% of the gap was closed in 12 instances, whereas

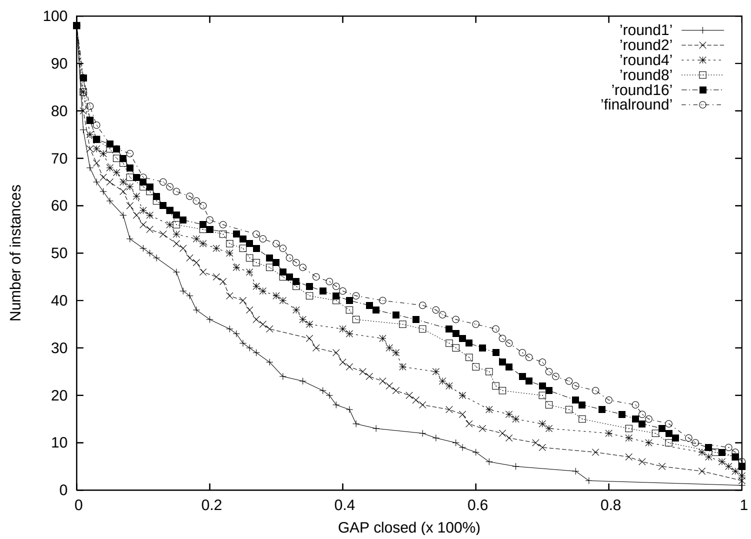

Figure 1: Multiple rounds of safe Gomory cuts.

four rounds of cuts closed 50% of the gap in 26 instances.

A similar chart is given in Figure 2, comparing the final bounds obtained with 128 rounds of safe cuts versus 128 rounds of the standard unsafe version of Gomory cuts. The two curves indicate that no significant loss in the bounds is incurred with the use of the accurate cutting planes.

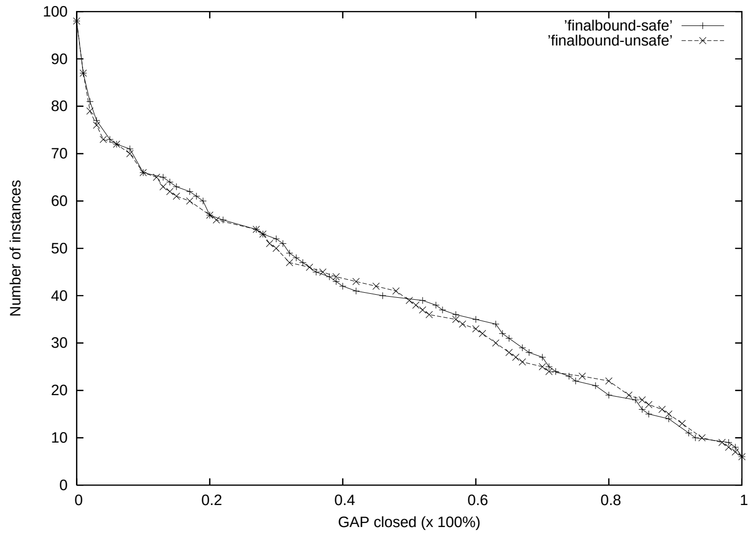

Figure 2: Safe vs. unsafe Gomory cuts.

References

Achterberg, T. 2007. *Constraint Integer Programming*. Ph.D. Thesis. Technischen Universtität Berlin, Berlin, Germany.

Achterberg, T., T. Koch, A. Martin. 2006. MIPLIB 2003. Op. Res. Letters **34**, 361–372.

Applegate, D., R. E. Bixby, V. Chvátal, W. Cook. 2006. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, New Jersey, USA.

Applegate D., W. Cook, S. Dash, D. Espinoza. 2007. Exact solutions to linear programming problems. Op. Res. Letters. To appear.

Balas, E., S. Ceria, G. Cornuéjols, N. Natraj. 1996. Gomory cuts revisited. Op. Res. Letters **19**, 1–9.

Balas, E., A. Saxena. 2006. Optimizing over the split closure. Math. Prog. To appear.

Bertsimas, D., R. Weismantel. 2005. *Optimization over Integers*. Dynamic Ideas, Belmont, Massachusetts, USA.

Bixby, R. E., S. Ceria, C. M. McZeal, M. W. P Savelsbergh. 1998. An updated mixed integer programming library: MIPLIB 3.0. Optima **58**, 12–15.

Bixby, R. E., M. Fenelon, Z. Gu, E. Rothberg, R. Wunderling. 2000. MIP: Theory and practice—closing the gap. M. J. D. Powell, S. Scholtes, eds. *System Modelling and Optimization: Methods, Theory and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands. 19–49.

Bixby, R. E., M. Fenelon, Z. Gu, E. Rothberg, R. Wunderling. 2004. Mixed-integer programming: A progress report. M. Grötschel, ed. *The Sharpest Cut: The Impact of Manfred Padberg and His Work*. SIAM, Philadelphia, Pennsylvania, USA. 309–325

Cornuéjols, G., Y. Li, D. Vandenbussche. 2003. K-cuts: A variation of Gomory mixed integer cuts from the LP tableau. INFORMS J. Comp. **15**, 385–396.

Dash, S., M. Goycoolea, O. Günlük. 2006. Two-step MIR inequalities for mixed-integer programs. Technical report. Available at Optimization Online.

Dash, S., O. Günlük, A. Lodi. 2007. On the MIR closure of polyhedra. *Integer Programming and Combinatorial Optimization* (IPCO XII). Lecture Notes in Computer Science **4513**. Springer, Heidelberg, Germany. 337-351.

Goldberg, D. 1991. What every computer scientist should know about floating-point arithmetic. ACM Comp. Surveys **23**, 5–48.

Gomory, R. E. 1960. An algorithm for the mixed integer problem. RM-2597. The Rand Corporation.

Goycoolea, M. 2006. *Cutting Planes for Large Mixed Integer Programming Models*. Ph.D. Thesis. Georgia Institute of Technology, Atlanta, Georgia, USA.

IEEE. 1985. *Standard for Binary Floating Point Arithmetic*, ANSI/IEEE Standard 754-1985.

ISO/IEC.1999. *Programming Languages–C*, ISO/IEC 9899:1999.

Marchand, M., L. A. Wolsey. 2001. Aggregation and mixed integer rounding to solve MIPs. Op. Res. **49**, 363–371.

Margot, F. 2007. Testing cut generators for mixed-integer linear programming. Tepper Working Paper 2007 E-43, Carnegie Mellon University, Pittsburgh, USA.

Martin, G. T. 1966. Solving the traveling salesman problem by integer linear programming. Technical Report, C-E-I-R, New York, USA.

Miliotis, P. 1978. Using cutting planes to solve the symmetric travelling salesman problem. Math. Prog. **15**, 177–188.

Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, USA.

Neumaier, A., O. Shcherbina. 2004. Safe bounds in linear and mixed-integer linear programming. Math. Prog. **99**, 283–296.

Reinelt, G. 1991. TSPLIB—a traveling salesman library. ORSA J. Comp. **3**, 376–384.

Wolsey, L. A. 1998. *Integer Programming*. John Wiley & Sons, New York, USA.