CORE DISCUSSION PAPER
9839

AGGREGATION and MIXED INTEGER ROUNDING to SOLVE MIPs

Hugues Marchand[1] and *Laurence A. Wolsey*[2]

June 1998

Abstract

A separation heuristic for mixed integer programs is presented that theoretically allows one to derive several families of "strong" valid inequalities for specific models and computationally gives results as good as or better than those obtained from several existing separation routines including flow cover and integer cover inequalities. The heuristic is based on aggregation of constraints of the original formulation and mixed integer rounding inequalities.

Keywords: mixed integer programming, cutting planes, Gomory mixed integer cuts.

[1]CORE, Université catholique de Louvain. E-mail hmarchand@core.ucl.ac.be
[2]CORE and INMA, Université catholique de Louvain. E-mail wolsey@core.ucl.ac.be
The first author was supported by a doctoral fellowship from Collège Interuniversitaire pour les Sciences du Management (CIM). This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsability is assumed by the authors.

1 Introduction

Gomory mixed integer cuts were proposed forty years ago, but, in spite of a finite convergence theorem, have apparently been little used in practice. More recently a variety of results and observations encourage the view that these inequalities or variants of these inequalities are "strong". Accepting this hypothesis, the question addressed here is whether and how such inequalities can be used effectively. Our conclusion is that using a constraint and bound aggregation procedure, and mixed integer rounding inequalities, it is possible to derive many strong valid inequalities that have been proposed for specific models in the literature, and also to obtain computational results as good as or better than those obtained with several existing separation routines such as those generating flow cover, and surrogate and integer knapsack inequalities.

The points suggesting that Gomory mixed integer cuts are strong include: i) the result that several general methods of generating inequalities for general mixed integer problems are equivalent (specifically disjunctive, split, Gomory mixed and mixed integer rounding inequalities) [8],[17], [18], ii) the result [1] that for 0-1 mixed integer programs variable by variable convexification leads to the convex hull, see also [12], iii) the observations of various authors [4], [8] that certain families of strong inequalities are disjunctive, split or MIR inequalities, iv) the recent computational results showing that lift-and-project (or disjunctive) cuts [2], and Gomory mixed integer cuts [3] can be effective when used correctly in a branch-and-cut framework and v) the results [14] showing that single node flow sets can be viewed and treated computationally as mixed knapsack sets, thereby simplifying the generation of flow cover inequalities.

The paper is self-contained. In Section 2 we derive the basic mixed integer rounding (MIR) inequality, and Gomory's mixed integer cut from scratch. In Section 3 we demonstrate point iii), showing that several families of strong valid inequalities are MIR inequalities. In Section 4 we define a mixed knapsack set, and the set of "complemented mixed integer rounding" (c-MIR) inequalities. We then define a separation heuristic for mixed integer sets, including an aggregation step, an active bound selection step leading to a mixed knapsack set, and finally a c-MIR separation heuristic for mixed knapsack sets involving complementation and scaling. Finally comparative computational results on a set of network design and unit commitment problems, as well as the mixed integer instances from the MIPLIB3.0 library, are presented.

2 The Mixed Integer Rounding and Gomory Mixed Integer Inequalities

To develop the standard procedures for generating mixed integer cuts, it suffices to study a two variable set

$$X = \{(x, y) \in R^1_+ \times Z^1 : y \leq b + x\}.$$

Note that the "disjunctive argument", used below to establish the validity of the valid inequality for X, is very simple: if the inequality $\pi x \leq \pi_0$ is valid for X^i for $i = 1, 2$, then it is valid for $X = X^1 \cup X^2$.

Proposition 1 *Let $f = b - \lfloor b \rfloor$. The inequality*

$$y \leq \lfloor b \rfloor + \frac{x}{1-f} \tag{1}$$

is valid for X.

Proof. Let $X^1 = X \cap \{(x, y) : y \leq \lfloor b \rfloor\}$ and $X^2 = X \cap \{(x, y) : y \geq \lfloor b \rfloor + 1\}$.

For $(x, y) \in X^1$, $y - \lfloor b \rfloor \leq 0$ and $0 \leq x$. Combining these two inequalities with weights $(1 - f)$ and 1 respectively gives $(y - \lfloor b \rfloor)(1 - f) \leq x$.

For $(x, y) \in X^2$, $-y + \lfloor b \rfloor + 1 \leq 0$ and $y - b \leq x$. Rewriting these inequalities as $-(y - \lfloor b \rfloor) \leq -1$ and $(y - \lfloor b \rfloor) \leq f + x$ and combining with weights f and 1 respectively gives $(y - \lfloor b \rfloor)(1 - f) \leq x$.

Now by the disjunctive argument, (1) is valid for $X^1 \cup X^2 = X$. ∎

We now use inequality (1) to derive the inequality used throughout this paper. Consider the single constraint mixed integer set

$$Y = \{(x, y) \in R^2_+ \times Z^{|N|}_+ : \sum_{j \in N} a_j y_j + x^+ \leq b + x^-\}.$$

Proposition 2 *Let $f = b - \lfloor b \rfloor$ and $f_j = a_j - \lfloor a_j \rfloor$ for $j \in N$. The inequality*

$$\sum_{j \in N} (\lfloor a_j \rfloor + \frac{(f_j - f)^+}{1-f}) y_j \leq \lfloor b \rfloor + \frac{x^-}{1-f} \tag{2}$$

is valid for Y.

Proof. Let $N_1 = \{j \in N : f_j \leq f\}$ and $N_2 = N \setminus N_1$. Let $w = \sum_{j \in N_1} \lfloor a_j \rfloor y_j + \sum_{j \in N_2} \lceil a_j \rceil y_j \in Z^1$. As $y \geq 0$, the inequality $w \leq b + x^- + \sum_{j \in N_2}(1 - f_j)y_j$ defines a relaxation of Y with $w \in Z^1$ and $x^- + \sum_{j \in N_2}(1 - f_j)y_j \geq 0$. Applying Proposition 1, we obtain that

$$w \leq \lfloor b \rfloor + \frac{x^- + \sum_{j \in N_2}(1 - f_j)y_j}{1 - f}$$

is valid for Y, which after substituting for w gives (2). ∎

We call (2) the *mixed integer rounding* (MIR) inequality, see [18].

Finally we show that the MIR inequality (2) is essentially just a more direct version of the Gomory mixed integer cut. Consider the set

$$Y^= = \{(x, y_0, y) \in R^2_+ \times Z^1 \times Z^{|N|}_+ : y_0 + \sum_{j \in N} a_j y_j + x^+ - x^- = b\},$$

which can be thought of as a row of an LP tableau with y_0 the basic variable, and x^+/x^- the sum of all the continuous variables with positive/negative coefficients respectively.

Proposition 3 *Let $N_1 = \{j \in N : f_j \leq f\}$ and $N_2 = N \setminus N_1$. The inequality*

$$\sum_{j \in N_1} f_j y_j + \sum_{j \in N_2} \frac{f(1 - f_j)}{f_j} y_j + x^+ + \frac{f}{1 - f} x^- \geq f \tag{3}$$

is valid for $Y^=$.

Proof. Relaxing the equality to an inequality $y_0 + \sum_{j \in N} a_j y_j + x^+ - x^- \leq b$, the MIR inequality of Proposition 2 gives

$$y_0 + \sum_{j \in N} (\lfloor a_j \rfloor + \frac{(f_j - f)^+}{1 - f}) y_j \leq \lfloor b \rfloor + \frac{x^-}{1 - f}$$

Substitution for y_0 leads to the inequality (3). ∎

Further discussion of the different ways to generate general mixed integer cuts can be found in [17], [18].

3 Many Strong Valid Inequalities are MIR

Various researchers have observed that certain "strong" valid inequalities derived by special arguments for structured mixed integer sets can also be derived as MIR inequalities. Here we present three examples of this situation.

3.1 Residual Capacity Inequalities

Let $X^F = \{(x, y) \in R^1_+ \times Z^1_+ : x \leq Cy, x \leq d\}$. Such sets arise in many network design models where x is the flow of a given commodity or set of commodities in an edge or across a cut-set, d is the demand in the commodity or set of commodities and capacity consisting of y multiples of C is installed on the edge or cut-set.

Proposition 4 *Let $\eta = \lceil \frac{d}{C} \rceil$ and $r = d - C(\eta - 1)$. Suppose that $0 < r < 1$. The residual capacity inequality [13] for X^F,*

$$x \leq d - r(\eta - y), \tag{4}$$

is a MIR inequality.

Proof. Set $s = d - x$, so that X^F can be rewritten as $-y \leq -\frac{d}{C} + \frac{s}{C}$ with $s \geq 0$ and $y \in Z^1$. Now the MIR inequality (2) takes the form $-y \leq -\lceil \frac{d}{C} \rceil + \frac{s}{Cf}$ where $f = \frac{d}{C} - \lfloor \frac{d}{C} \rfloor$. Taking $r = Cf = d - C\lfloor \frac{d}{C} \rfloor = d - C(\eta - 1)$ and substituting for s gives (4). ∎

Various generalizations of the set X^F have been studied. Two examples are

$$X^F_1 = \{(x, y) \in R^{|N|}_+ \times Z^1_+ : \sum_{j \in N} x_j \leq Cy, x_j \leq u_j \text{ for } j \in N\}$$

which is a multicommodity version of the set X^F, see [13], and

$$X^F_2 = \{(x, y) \in R^1_+ \times Z^{|N|}_+ : x \leq \sum_{j \in N} C_j y_j, x \leq d\}$$

where the C_i's represent several technologies with different capacities to send flow on an edge or across a cutset, and $C_1 | C_2 | \ldots | C_n$, see [21]. In both these cases all the non-trivial facet-defining inequalities are MIR inequalities.

In [4] it is observed that several other network design inequalities are MIR inequalities, and recently a family of inequalities for a multicommodity single path network model have also been shown to be dominated by MIR inequalities [6].

3.2 Mixed Cover Inequalities

Consider the mixed 0-1 knapsack set

$$X^B = \{(y,s) : B^{|N|} \times R^1_+ : \sum_{j \in N} a_j y_j \leq b + s\}$$

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

Let $C \subseteq N$ be a cover with excess λ, i.e. $\sum_{j \in C} a_j = b + \lambda$ with $\bar{a} > \lambda > 0$ where $\bar{a} = \max_{j \in C} a_j$. Let $E(C) = C \cup \{j \in N \setminus C : a_j \geq \bar{a}\}$ be the extension of C.

Proposition 5 *The mixed cover inequality for X^B,*

$$\sum_{j \in E(C)} \min(a_j, \lambda) y_j \leq -\lambda + \sum_{j \in C} \min(a_j, \lambda) + s, \tag{5}$$

is a MIR inequality.

Proof. Complementing y_j for $j \in C$ and dividing the original knapsack constraint by $\overline{a}$, we obtain,

$$\sum_{j \in N \setminus C} \frac{a_j}{\overline{a}} y_j - \sum_{j \in C} \frac{a_j}{\overline{a}} \overline{y}_j \leq -\frac{\lambda}{\overline{a}} + \frac{s}{\overline{a}}$$

where $\overline{y}_j = 1 - y_j$. Using $y_j \geq 0$ for $j \in N \setminus C$ and $a_j \geq \overline{a}$ for $j \in E(C) \setminus C$, we obtain the relaxation,

$$\{(y,s) \in B^{|N|} \times R^1_+ \ : \ \sum_{j \in E(C) \setminus C} y_j - \sum_{j \in C} \frac{a_j}{\overline{a}} \overline{y}_j \leq -\frac{\lambda}{\overline{a}} + \frac{s}{\overline{a}}\}.$$

Using $1 - f = \frac{\lambda}{\overline{a}}$ and $f_j = 1 - \frac{a_j}{\overline{a}}$ for $j \in C$, the MIR inequality for this relaxation is

$$\sum_{j \in E(C) \setminus C} y_j - \sum_{j \in C} \frac{\min(a_j, \lambda)}{\lambda} \overline{y}_j \leq -1 + \frac{s}{\lambda}$$

or equivalently, back in the original space of variables, the mixed cover inequality (5). ∎

Adding $0 \leq \sum_{j \in C}(\lambda - a_j)^+(1 - y_j)$ to (5) and dividing by λ gives

$$\sum_{j \in E(C)} y_j \leq |C| - 1 + \frac{s}{\lambda}$$

which is a generalization of the cover inequality for the mixed knapsack set [7].

Flow cover inequalities [19] are obtained from (5) using the mixed knapsack relaxation of the flow models introduced in [14]. As a consequence, flow cover inequalities are also MIR inequalities. In [15] it is shown that several other classes of flow cover inequalities are also MIR inequalities.

3.3 Weight Inequalities

Consider a general mixed knapsack set,

$$\begin{aligned} X^W = \{(x, y) \in R_+^{|P|} \times Z_+^{|N|} : \textstyle\sum_{j \in P} a_j x_j + \sum_{j \in N} a_j y_j \leq b, \\ x_j \leq u_j \text{ for } j \in P,\ y_j \leq h_j \text{ for } j \in N\}, \end{aligned}$$

as studied in [16].

Proposition 6 *Let $T \subseteq N, Q \subseteq P$ with $\mu = b - \sum_{j \in Q} a_j u_j - \sum_{j \in T} a_j h_j > 0$, and $\bar{a} = \max_{j \in N \setminus T} a_j > \mu$. The weight inequality [16] for X^W*

$$\sum_{j \in Q} a_j x_j + \sum_{j \in T} a_j y_j + \sum_{j \in N \setminus T} (a_j - \mu)^+ y_j \leq b - \mu \tag{6}$$

is a MIR inequality.

Proof. Complementing the variables in Q and T gives

$$\sum_{j \in P \setminus Q} a_j x_j + \sum_{j \in N \setminus T} a_j y_j \leq \mu + \sum_{j \in Q} a_j \bar{x}_j + \sum_{j \in T} a_j \bar{y}_j$$

where $\bar{x}_j = u_j - x_j$ for $j \in Q$, and $\bar{y}_j = h_j - y_j$ for $j \in T$. Relaxing this constraint leads to the following relaxation of set X^W

$$\{(y, s) \in Z^{|N \setminus T|} \times R_+^1 \ : \sum_{j \in N \setminus T} \frac{a_j}{\bar{a}} y_j \leq \frac{\mu}{\bar{a}} + \frac{s}{\bar{a}}\}$$

where $s = \sum_{j\in Q} a_j\bar{x}_j + \sum_{j\in T} a_j\bar{y}_j \geq 0$. The MIR inequality for this relaxation is

$$\sum_{j\in N\setminus T} \frac{(a_j-\mu)^+}{(\bar{a}-\mu)} y_j \leq \frac{s}{(\bar{a}-\mu)}$$

Multiplying by $\bar{a}-\mu$ and substituting for s gives the inequality (6). ∎

In addition it is shown in [16] that, in certain cases, all the nontrivial facet-defining inequalities are weight inequalities.

4 Mixed Knapsack Sets and c-MIR Inequalities

Consider the set

$$X^{MK} = \{(y,s) \in Z_+^{|N|} \times R_+^1 : \sum_{j\in N} a_j y_j \leq b + s, y_j \leq u_j \text{ for } j \in N\},$$

called a mixed knapsack set.

Let (T,C) be a partition of N and $\delta > 0$. A *complemented MIR* (c-MIR) inequality for X^{MK} associated to (T,C) and δ is obtained by complementing variables in C, and dividing by δ before generating the MIR inequality. Specifically this gives the set

$$\{(y,s) \in Z^{|N|} \times R_+^1 \ : \ \sum_{j\in T} \frac{a_j}{\delta} y_j + \sum_{j\in C} \frac{-a_j}{\delta} \bar{y}_j \leq \frac{b - \sum_{j\in C} a_j u_j}{\delta} + \frac{s}{\delta}\}$$

with c-MIR inequality

$$\sum_{j\in T} G(\frac{a_j}{\delta}) y_j + \sum_{j\in C} G(\frac{-a_j}{\delta})(u_j - y_j) \leq \lfloor\beta\rfloor + \frac{s}{\delta(1-f)}$$

where $\beta = (b - \sum_{j\in C} a_j u_j)/\delta, f = \beta - \lfloor\beta\rfloor$ and $G(d) = (\lfloor d\rfloor + \frac{(f_d-f)^+}{1-f})$ with $f_d = d - \lfloor d\rfloor$.

Clearly each of the examples in Section 3 can be viewed as the construction of a relaxation of the form X^{MK}, and then the generation of a c-MIR inequality for X^{MK}. This is the procedure that we now try to make automatic.

5 A Separation Procedure for MIPs

The procedure consists of three steps, each of them heuristic:

i) **Aggregation** in which one or more rows of the matrix are combined so as to obtain a single mixed integer constraint,

ii) **Bound Substitution** in which the slack variables of the simple or variable lower and upper bounds constraints are introduced, and the resulting set is a mixed knapsack set of the form X^{MK},

iii) **c-MIR Separation** for the mixed knapsack set X^{MK}.

Specifically we suppose that the original problem constraints have been classified into (a) general mixed integer rows taking the form

$$\sum_{j \in P} a^i_j x_j + \sum_{j \in N} g^i_j y_j = b_i \text{ for } i \in M$$

after addition of slack variables to create equalities if necessary, (b) simple or variable lower and upper bound constraints

$$l_j y_j \leq x_j \leq u_j y_j \text{ or } l_j \leq x_j \leq u_j$$

and (c) other constraints that we ignore. For simplicity of notation, we assume below that all bound constraints are variable lower and upper bounds. Also a fractional point (x^*, y^*) is given. We now specify the three heuristic steps of the procedure.

The Aggregation Heuristic.

Suppose that a subset $S \subseteq M$ of the rows have been combined to form a single mixed integer constraint

$$\sum_{j \in P} \alpha_j x_j + \sum_{j \in N} \gamma_j y_j = \beta.$$

The bound substitution and c-MIR separation procedures are called for this constraint. If no violated inequality is found, we try to add in another constraint $i \in M \setminus S$ as follows. We essentially use the strategy introduced in [22] to create paths in fixed charge networks. Let

$$P^* = \{k \in P : \alpha_k \neq 0,\ l_k y^*_k < x^*_k < u_k y^*_k \text{ and } \exists\, r \in M \setminus S \text{ with } a^r_k \neq 0\}$$

be the set of variables that are candidates to create a link to a new constraint. If P^* is empty, stop. Otherwise choose the variable x_κ, $\kappa \in P^*$, that is furthest from its bounds, i.e. choose κ such that $\kappa = \arg\max\{\Delta_k : k \in P^*\}$ and $\Delta_k = \min\{x_k^* - l_k y_k^*, u_k y_k^* - x_k^*\}$. Choose also a row $r \in M \setminus S$ with $a_\kappa^r \neq 0$.

The current mixed integer row and row r are now aggregated in such a way that the coefficient of x_κ becomes zero. So the new aggregated constraint has

$$\alpha_j \leftarrow \alpha_j - \frac{\alpha_\kappa}{a_\kappa^r} a_j^r \text{ for } j \in P,\ \gamma_j \leftarrow \gamma_j - \frac{\alpha_\kappa}{a_\kappa^r} g_j^r \text{ for } j \in N,$$

$$\beta \leftarrow \beta - \frac{\alpha_\kappa}{a_\kappa^r} b_r \text{ and } S \leftarrow S \cup \{r\}.$$

The aggregation procedure is stopped when $|S|$ exceeds some specified value MAXAGGR.

The Bound Substitution Heuristic.

Consider the mixed integer constraint

$$\sum_{j \in P} \alpha_j x_j + \sum_{j \in N} \gamma_j y_j = \beta$$

derived by aggregation together with the variable bound constraints on the continuous variables x_j for $j \in P$. For each such variable, either the substitution $x_j = l_j y_j + t_j$, or the substitution $x_j = u_j y_j - t_j$ is selected, leading to a new constraint

$$\sum_{j \in N} \gamma_j' y_j + \sum_{j \in P} \delta_j' t_j = \beta'.$$

Letting $s = \sum_{j \in P : \delta_j' < 0} (-\delta_j') t_j \geq 0$, we obtain the mixed knapsack set

$$X^{MK} = \{(y, s) \in Z_+^{|N|} \times R_+^1 : \sum_{j \in N} \gamma_j' y_j \leq \beta' + s, y_j \leq u_j \text{ for } j \in N\}.$$

Different criteria can be used to decide how to eliminate each continuous variable by introducing the slack variable associated with either its variable lower or upper bound constraint:

a) minimize the difference between each continuous variable and its bound, i.e. substitute $x_j = l_j y_j + t_j$ if and only if $x_j^* - l_j y_j^* \leq u_j y_j^* - x_j^*$,

b) minimize the value of $s^* = \sum_{j\in P:\delta'_j<0}(-\delta'_j)t^*_j$. If $u_j = \infty$ or $x^*_j = l_j y^*_j$, substitute $x_j = l_j y_j + t_j$. If $x^*_j = u_j y^*_j$, substitute $x_j = u_j y_j - t_j$. Otherwise, if $\alpha_j < 0$, substitute $x_j = u_j y_j - t_j$ and, if $\alpha_j > 0$, substitute $x_j = l_j y_j + t_j$,

c) the opposite of b), minimize the value of $\sum_{j\in P:\delta'_j>0}(\delta'_j)t^*_j$.

The c-MIR Separation Heuristic.

Consider a mixed knapsack set

$$X^{MK} = \{(y,s) \in Z_+^{|N|} \times R_+^1 : \sum_{j\in N} a_j y_j \leq b + s, y_j \leq u_j \text{ for } j \in N\}$$

and a fractional point (y^*, s^*).

To choose the set $C \subseteq N$ and the divisor δ to be used, we first take $C = \{j \in N : y^*_j \geq \frac{u_j}{2}\}$ with $T = N \setminus C$. As candidate set for δ, we take $\{a_j : j \in N \text{ and } 0 < y^*_j < u_j\}$. Testing each candidate in turn, we take the MIR inequality with the greatest violation.

To try to increase the violation, the δ values are divided by 2,4 and 8 respectively, and the c-MIR inequality with the largest violation is selected. Finally for the chosen value of δ, we check if the violation is increased by successively complementing each variable lying strictly between its bounds, ordered by nondecreasing values of $|y^*_j - \frac{u_j}{2}|$.

6 Computational Results

The c-MIR separation heuristic has been inserted into the branch-and-cut system bc-opt [9]. We have compared the default version of bc-opt which contains a variety of mixed integer separation routines (flow cover, surrogate knapsack, integer knapsack, Gomory mixed integer cuts) and a version with just the c-MIR routine. The test set consists of a set of unit commitment and network design problems, and the mixed integer instances in the MIPLIB3.0 library.

The test have been carried out with Version 10.28 of bc-opt, running on a Pentium 2-233 with 64M of RAM, using the default preprocessing of XPRESS, with a best bound strategy for 255 nodes and the XPRESS default strategy thereafter. The node limit is set to be 300000 nodes, time limit to 5400 seconds, and MAXAGGR = 6 for the generation of c-MIR inequalities. Cutting planes are added at the top node only except for the network design problem where they are added at each node of the branch-and-bound tree.

6.1 The Unit Commitment Problem

The objective in these instances [20] is to minimise electricity production costs given the production sites, the operating characteristics of the generators on these sites, the transportation links between sites, and the estimation of the demand for electricity for each hour of the day. This model contains general integer variables which represent the numbers of generators of the same type in use. Mixed knapsack constraints modelling the satisfaction of the electricity demand at each site as well as variable lower and upper bounds on the production variables are also present.

We have six instances of this model corresponding to three different scenarios and two different formulations. The instances gesa_io for $i = 1, 2, 3$ use the original formulation. In the instances gesa_i for $i = 1, 2, 3$, redundant aggregated (over sites) demand constraints have been added. In Table 1, the size of the instances is indicated. Columns 1-4 contain the number of rows, binary, integer and continuous variables respectively. Column 5 shows the initial lp-value and Column 6 the optimal ip-value. In all the tables below, N means that node limit is reached, T means that time limit is reached and M means that memory problems occurred (too many active nodes). Time is measured in seconds. Only gesa_3 and gesa_3o can be solved in less than 5400 seconds by bc-opt using a pure branch-and-bound strategy.

instance	m	B	I	C	lp	ip
gesa_1	1468	240	168	816	22052765	22311872
gesa_2	1345	240	168	768	25492512	25779856
gesa_3	1297	216	168	768	27846437	27911042
gesa_1o	1249	384	336	504	21826559	22311872
gesa_2o	1201	384	336	456	25476489	25779856
gesa_3o	1153	336	312	432	27833632	27911042

Table 1: Gesa: description of the six instances.

In Table 2, we compare results obtained by bc-opt using the default cutting planes with those obtained by bc-opt using c-MIR inequalities. In this experiment, cutting planes are added at the top node only. xlp, time and nodes are respectively the lp-value after adding cuts, the time and the number of nodes taken by the cut-and-branch to solve the different instances of the model.

We observe that bc-opt with the default cutting planes is able to solve

	with default cutting planes			with c-MIR inequalities		
instance	xlp	time	nodes	xlp	time	nodes
gesa_1	22226863	61	2663	22298415	14	15
gesa_2	25705338	107	7020	25776094	13	72
gesa_3	27929735	10	224	27945861	7	156
gesa_1o	21879640	M	M	22250970	31	533
gesa_2o	25708669	4026	241414	25757089	14	168
gesa_3o	27916557	10	437	27943158	8	171

Table 2: Gesa: result cut-and-branch.

efficiently to optimality the instances of the model with the redundant constraints but not the original ones: gesa1o is not solved and gesa2o is solved but enumerating 241414 nodes. bc-opt with only c-MIR inequalities solves both types of instances. Apparently, the right constraints are combined together by the aggregation heuristic.

6.2 The Network Design Problem

The problem here is to determine the cost-optimal configuration of a nationwide operating corporation's telecommunication service network based on given demands of voice and data transmission [6]. This model can be seen as a special case of a minimum-cost multicommodity network flow problem where each commodity has to be routed on a single path either directly from its origin to its destination or by transiting through a certain number of hub-nodes. As a consequence, this model contains only integer variables: binary variables that represent decisions about using arcs to route the commodities and general integer variables to represent the capacities installed on the arcs of the network.

In Table 3, the four instances of this model are briefly described. Columns 1-2 contain the number of nodes and hub nodes in the network, Column 3 contains the number of commodities, Columns 4-6 contain the number of rows, binary and integer variables, Column 7, the initial lp-value and Column 8 the optimal ip-value. Only b_20 can be solved in less than 5400 seconds by bc-opt using a pure branch-and-bound strategy.

In the three first columns of Table 4, we report results obtained by bc-opt using the default cutting planes and some problem specific cuts [6] at each node of the tree. In the last three, the results with the c-MIR inequalities

instance	n	h	c	m	B	I	lp	ip
b_20	20	8	24	477	1815	37	13644	17262
b_28	28	8	32	685	2459	38	17320	21107
b_32	32	10	49	1109	5644	58	21509	26689
b_54	54	10	74	1849	8462	68	29256	34406

Table 3: BASF: description of the four instances.

are reported. Again xlp, time and nodes are respectively the lp-value after adding cuts, the time and the number of nodes taken by the branch-and-cut to solve the different instances of the model.

	with default + specific cuts			with c-MIR inequalities		
instance	xlp	time	nodes	xlp	time	nodes
b_20	15977	269	1147	16423	167	183
b_28	19858	562	1360	20213	344	287
b_32	25669	T	T	26036	5129	1213
b_54	33505	T	T	33817	3909	314

Table 4: BASF: result branch-and-cut.

We observe here that c-MIR inequalities are only generated on single constraints of the initial formulation of the model because this model does not contain continuous variables. Therefore no aggregation of constraints is performed by the cutting plane procedure. However generating these c-MIR inequalities suffices to solve all the instances of the problem to optimality without using problem specific cuts in less than 5400 seconds.

We observe also that generating cutting planes at each node of the branch-and-bound tree is required to solve these problems to optimality.

6.3 The MIPLIB3.0 Test Set

The third test set is composed of all the mixed integer problems of MIPLIB3.0 [5] containing both integer and continuous variables except for arki001, bell5, danoint, blend2, pk1, noswot, rout, dano3mip, qiu and dsbmip. These ten problems are discarded because neither the default cutting planes of bc-opt nor the c-MIR inequalities are effective on them. We also do not report the results obtained on the gesa problems included in MIPLIB3.0

which have been treated earlier in this section.

In Table 5, we compare the results obtained by bc-opt solving these problems with a cut-and-branch approach using the default cutting planes (see the three first columns) and using c-MIR inequalities (see the three last columns). xlp, time and nodes are respectively the lp-value after adding cuts, the time and the number of nodes taken by the cut-and-branch to solve the problem.

	with default cuts			with c-MIR inequalities		
instance	xlp	time	nodes	xlp	time	nodes
bell3a	873929	482	100467	873351	604	123804
egout	568	1	1	560	0	40
fixnet6	3687	10	135	3840	34	333
modglob	20694470	80	16322	20726335	2	75
pp08a	7108	24	2633	7087	50	2970
pp08aCUTS	7108	25	2532	7190	41	1907
rgn	65	8	2256	82.20	2	6
set1ch	54517	5	117	51826	T	T
vpm1	19.5	2	312	20	2	1
vpm2	12.21	1804	292999	12.77	184	22250
gen	112233	1	7	112313	1	1
khb	106740497	2	27	106856260	5	24
dcmulti	184988	6	592	184283	6	853
flugpl	1172656	11	5199	1167875	11	5743
rentacar	29363157	33	24	29214734	37	33
misc06	12848	3	151	12841	7	780
mod011	-62046188	T	T	-58316650	1705	6119

Table 5: MIPLIB 3.0: result cut-and-branch.

On the one hand, we observe that bc-opt with c-MIR inequalities cannot solve set1ch. Path inequalities seem to be required to solve this problem to optimality (see [9]). Moreover no c-MIR inequalities are generated on misc06. On the other hand, modglob, rgn, vpm1, vpm2 and mod011 are solved more easily using c-MIR inequalities than using the default cutting planes. Both cut-and-branch approaches have difficulties in solving bell3a and bell5, while these instances are solved rapidly in [16].

Various other tests have been carried out on the MIPLIB3.0. For the instances with integer and not just 0-1 variables, a comparison was made between c-MIR inequalities alone, and c-MIR inequalities with integer knapsack inequalities [7]. The results were comparable, suggesting that, as in the network design instances, c-MIR inequalities could be used without these knapsack inequalities. However, for the 0-1 problems, and especially the 0-1 problems with GUBs such as mitre, p0548 and p2756, it is important to keep specialised routines generating lifted cover and GUB cover inequalities. For the mixed integer instances in Table 5, we have looked at the effect of using Gomory mixed integer cuts (of the optimal tableau) [18] after having generated c-MIR inequalities. The effect does not appear to be significant in reducing the running time and the size of the branch-and-bound tree.

These results suggest that a default branch-and-cut code should include a c-MIR separation routine together with separation routines for lifted cover and GUB cover inequalities. This observation needs to be validated on a larger set of problems.

References

[1] E. Balas, Disjunctive Programs: Cutting Planes from Logical Conditions, in O.L. Mangasarian et al., eds., *Nonlinear Programing, Vol. 2* Academic Press pp 279-312 (1975).

[2] E. Balas, S. Ceria and G. Cornuéjols, Mixed 0–1 Programming by Lift-and-Project in a Branch-and-cut Framework, *Management Science* **42**, 1229-1246 (1996).

[3] E. Balas, S. Ceria, G. Cornuéjols and G. Natraj, Gomory Cuts Revisited, *Operations Research Letters* **19**, 1-9 (1996).

[4] D. Bienstock and O. Günlük, Capacitated Network Design - Polyhedral Structure and Computation, *ORSA Journal on Computing* **8**, 243-259 (1996).

[5] R.E. Bixby, S. Ceria, C.M. McZeal and M.W.P. Savelsbergh, An Updated Mixed Integer Programming Library: MIPLIB 3.0, text and problem available at {http://www.caam.rice.edu/bixby/miplib/ miplib.html}.

[6] B. Brockmüller, O. Günlük and L.A. Wolsey, Designing Private Line Networks, CORE, Université catholique de Louvain, revised version of CORE DP9647 (1998).

[7] S. Ceria, C. Cordier, H. Marchand and Laurence A. Wolsey, Cutting Planes for Integer Programs with General Integer Variables, *Mathematical Programming* **81**, 201-214 (1998).

[8] W.J. Cook, R. Kannan and A. Schrijver, Chvátal Closures for Mixed Integer Programming Problems, *Mathematical Programming* **47**, 155-174 (1990).

[9] C. Cordier, H. Marchand, R. Laundy and L.A. Wolsey, $bc - opt$: a Branch-and-Cut Code for Mixed Integer Programs, CORE Discussion Paper 9778, Université Catholique de Louvain, October 1997

[10] R. E. Gomory, Solving Linear Programming Problems in Integers, in *Combinatorial Analysis*, R. E. Bellman and M. Hall, Jr., eds., American Mathematical Society, (1960), pp. 211-216.

[11] R.E. Gomory, An Algorithm for the Mixed Integer Problem, RM-2597 The Rand Corporation (1960).

[12] L. Lovász and A. Schrijver, Cones of Matrices and Set Functions and 0-1 Optimization, *SIAM Journal on Optimization* **1**, 166-190 (1991).

[13] T.L. Magnanti, P. Mirchandani and R. Vachani, The Convex Hull of Two Core Capacitated Network Design Problems, *Mathematical Programming* **60**, 233-250 (1993).

[14] H. Marchand and L.A. Wolsey, The 0-1 Knapsack Problem with a Single Continuous Variable, CORE DP 9720, Louvain-la-Neuve (1997).

[15] H. Marchand, A Study of the Mixed Knapsack Set and its Use to Solve Mixed Integer Programs, PhD Thesis, Faculté des Sciences Appliquées, Université catholique de Louvain, Louvain-la-Neuve (July 1998).

[16] A. Martin and R.Weismantel, Contributions to General Mixed Integer Knapsack Problems, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Preprint, SC 97-35 (1997).

[17] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York (1988).

[18] G.L. Nemhauser and L.A. Wolsey, A Recursive Procedure for Generating all Cuts for 0-1 Mixed Integer Programs, *Mathematical Programming* **46**, 379-390 (1990).

[19] M.W. Padberg, T.J. Van Roy and L.A. Wolsey, Valid Linear Inequalities for Fixed Charge Problems, *Operations Research* **33**, 842-861 (1985).

[20] Pamips, Esprit Project 8755, Public Report Ref. DR4.3.5/I, 31/1/95.

[21] Y. Pochet and L.A. Wolsey, Integer Knapsacks and Flow Covers with Divisible Coeffcients: Polyhedra, Optimization and Separation, *Discrete Applied Mathematics* **59**, 57-74 (1995).

[22] T.J. Van Roy and L.A. Wolsey, Solving Mixed 0-1 Problems by Automatic Reformulation, *Operations Research* **35**, 45-57 (1987).