Math Meth Oper Res (2001) 53:173–203

Mathematical Methods
of Operations Research

Classical cuts for mixed-integer programming and branch-and-cut*

Manfred Padberg

17, rue Vendôme, 13007 Marseille, France

Manuscript received: February 2000/Final version received: November 2000

Abstract. We review classical valid linear inequalities for mixed-integer programming, i.e., Gomory's fractional and mixed-integer cuts, and discuss their use in branch-and-cut. In particular, a generalization of the recent mixed-integer rounding (MIR) inequality and a sufficient condition for the global validity of classical cuts after branching has occurred are derived.

Key words: Mixed-integer programming, cutting planes, Gomory cuts, branch-and-cut

We consider mixed-integer programming problems of the form

$$\max\{\mathbf{cx} + \mathbf{dy} : \mathbf{Ax} + \mathbf{Dy} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0} \text{ and integer, } \mathbf{y} \ge \mathbf{0}\}, \qquad \text{(MIP)}$$

where $\mathbf{A}$ is a $m \times n$ and $\mathbf{D}$ a $m \times p$ matrix and all data are assumed to be rational. $\mathbf{x} \in \mathbb{Z}^n_+$ are the *integer* variables, $\mathbf{y} \in \mathbb{R}^p_+$ the *flow* or *continuous* variables of (MIP). We denote by

$$P(\mathbf{A}, \mathbf{D}, \mathbf{b}) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^p : \mathbf{Ax} + \mathbf{Dy} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0}, \mathbf{y} \ge \mathbf{0}\}$$

the polyhedron of the linear programming (LP) relaxation

$$\max\{\mathbf{cx} + \mathbf{dy} : \mathbf{Ax} + \mathbf{Dy} \le \mathbf{b}\ \mathbf{x} \ge \mathbf{0}, \mathbf{y} \ge \mathbf{0}\} \qquad (\text{MIP}_{LP})$$

of (MIP). The *convex hull* of the discrete-mixed set DM

$$DM = P(\mathbf{A}, \mathbf{D}, \mathbf{b}) \cap (\mathbb{Z}^n \times \mathbb{R}^p)$$

* Work supported in part by a grant from the Office of Naval Research (N00014-96-0327).

over which we wish to optimize is a polyhedron $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ in $\mathbb{R}^{n+p}$ because the data are by assumption rational, see e.g. [65], point 10.2(a), for a proof. $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ thus possesses a(n *ideal*) linear description and the existence of finite solution methods for (MIP) follows. The traditional "cutting plane" methods of the 1950's and 1960's largely ignored these mathematical underpinnings (e.g. the polyhedrality of $conv(DM)$ was established by R. R. Meyer [55] in full generality only in 1974). Rather they addressed the question of solving (MIP) by way of "valid inequalities" and "cutting planes" in a fairly direct, algorithmic way.

Definition 1. (i) *An inequality* $\mathbf{fx} + \mathbf{gy} \le f_0$ *is a* valid inequality *for (MIP) if* $\mathbf{fx} + \mathbf{gy} \le f_0$ *for all* $(\mathbf{x}, \mathbf{y}) \in DM$.

(ii) *A valid inequality* $\mathbf{fx} + \mathbf{gy} \le f_0$ *for (MIP) is a* cut *(or* cutting plane*) for (MIP) if*

$$P(\mathbf{A}, \mathbf{D}, \mathbf{b}) \cap \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : \mathbf{fx} + \mathbf{gy} \le f_0\} \subset P(\mathbf{A}, \mathbf{D}, \mathbf{b}),$$

where the containment is proper.

Let for simplicity $P_{LP_0} = P(\mathbf{A}, \mathbf{D}, \mathbf{b})$, $\mathbf{cx}^0 + \mathbf{dy}^0 = \max\{\mathbf{cx} + \mathbf{dy} : (\mathbf{x}, \mathbf{y}) \in P_{LP_0}\}$ and $\mathscr{F}$ be a family of cuts for (MIP) such that $\mathbf{fx}^0 + \mathbf{gy}^0 > f_0$ for all $(\mathbf{f}, \mathbf{g}, f_0) \in \mathscr{F}$. Then

$$P_{LP_1} = P_{LP_0} \cap \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : \mathbf{fx} + \mathbf{gy} \le f_0 \quad \text{for all } (\mathbf{f}, \mathbf{g}, f_0) \in \mathscr{F}\} \subset P_{LP_0},$$

unless $P_{LP_0} = \emptyset$, i.e., unless $z_0 = \mathbf{cx}^0 + \mathbf{dy}^0 = -\infty$, or $z_0 = +\infty$. In those two cases we can stop: (MIP) either has no feasible solution or an unbounded optimal solution. Otherwise,

$$z_1 = \max\{\mathbf{cx} + \mathbf{dy} : (\mathbf{x}, \mathbf{y}) \in P_{LP_1}\} \le z_0 = \max\{\mathbf{cx} + \mathbf{dy} : (\mathbf{x}, \mathbf{y}) \in P_{LP_0}\}.$$

Moreover, since every $(\mathbf{f}, \mathbf{g}, f_0) \in \mathscr{F}$ defines a valid inequality for (MIP)

$$DM \subseteq P_{LP_1} \subset P_{LP_0},$$

and we can iterate. Doing so generates a sequence of polyhedra satisfying

$$P_{LP_0} \supset P_{LP_1} \supset \cdots \supset P_{LP_\ell} \supset \cdots \supset P_{LP_k} \supseteq \cdots \supseteq DM$$

such that $z_{\ell+1} = \mathbf{cx}^{\ell+1} + \mathbf{dy}^{\ell+1} \le z_\ell = \mathbf{cx}^\ell + \mathbf{dy}^\ell$ for the optimal solutions $(\mathbf{x}^\ell, \mathbf{y}^\ell)$ to the linear program

$$\max\{\mathbf{cx} + \mathbf{dy} : (\mathbf{x}, \mathbf{y}) \in P_{LP_\ell}\}, \qquad (\mathrm{LP}_\ell)$$

where $\ell \ge 0$ and we stop if $\mathbf{x}^\ell \in \mathbb{Z}^n$ or $z_\ell = -\infty$, i.e., $P_{LP_\ell} = \emptyset$. Two basic questions ensue:

1. *Is it always possible to find* one or several cuts *with the desired property of cutting off the LP optimum* $(\mathbf{x}^\ell, \mathbf{y}^\ell)$ *if* $\mathbf{x}^\ell \notin \mathbb{Z}^n$*?*
2. *Does there exist a cut generation mechanism which guarantees* finite convergence *of the algorithm, i.e., does there exist a finite* ℓ *such that the optimal solution* $(\mathbf{x}^\ell, \mathbf{y}^\ell)$ *to the linear program* (LP_ℓ) *satisfies* $\mathbf{x}^\ell \in \mathbb{Z}^n$*?*

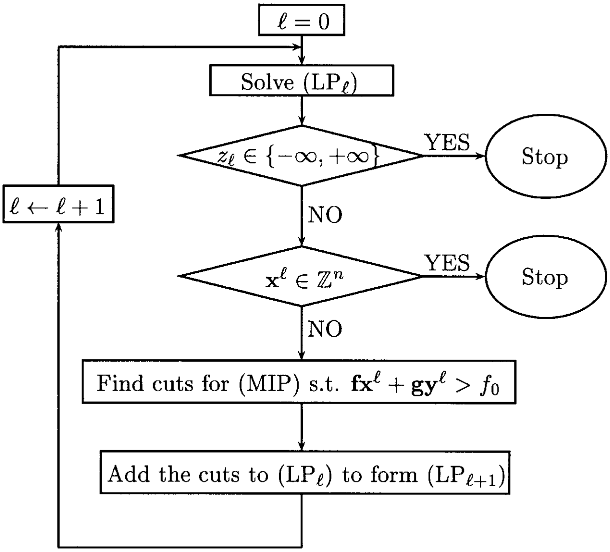

Fig. 1. Classical cutting plane algorithm A for MIP.

The answer to the purely *existential* question of a finite cutting plane algorithm for (MIP) along the lines of the flowchart of Figure 1 is positive and follows from the polyhedrality of the convex hull of DM: since $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ is a polyhedron in $\mathbb{R}^{n+p}$ it possesses a linear description by way of *finitely* many linear inequalities. It thus suffices to look for valid equations and facet-defining inequalities of $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ that are violated by the current LP optimum $(\mathbf{x}^\ell, \mathbf{y}^\ell)$ if $\mathbf{x}^\ell \notin \mathbb{Z}^n$. Of course, such a linear description may not be known, which leaves the algorithmics of numerical computation, i.e., the prescription of a cut generation mechanism, wide open.

1 Rules for deriving valid inequalities for MIP

We list next a number of well-known rules that can be used to derive a valid inequality (VI) for (MIP) from the original formulation of the problem. We write $\xi = (\mathbf{x}, \mathbf{y})$ where $\mathbf{x} \in \mathbb{Z}^n$, $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{y} \in \mathbb{R}^p$, $\mathbf{y} \geq \mathbf{0}$ and thus $\xi \geq \mathbf{0}$. Any combination of the following rules in arbitrary order of execution "works", i.e., produces a VI for (MIP) from other VIs of (MIP). Below we assume throughout that all variables must be nonnegative.

Rule I. Nonnegative combinations of VIs give VIs:

If $\mathbf{f}\xi \leq f_0$ *and* $\mathbf{h}\xi \leq h_0$ *are VIs, then* $(\lambda\mathbf{f} + \mu\mathbf{h})\xi \leq \lambda f_0 + \mu h_0$ *is a VI for all* $\lambda \geq 0$ *and* $\mu \geq 0$.

Example 1. If the inequalities

$$2x_1 - x_2 + 12x_3 - y_1 \le -2 \tag{1}$$

$$4x_1 + x_2 + y_1 \le 7 \tag{2}$$

are valid inequalities for some (MIP), then so is the inequality (use $\lambda = \mu = 1$)

$$6x_1 + 12x_3 \le 5. \tag{3}$$

Rule II. "Weakening" a VI gives a VI:

If $\mathbf{f}\xi \le f_0$ *is a VI, then* $\mathbf{f}^-\xi \le f_0$ *is a VI where* $\mathbf{f}^- = \min\{\mathbf{0}, \mathbf{f}\}$. *Likewise, if* $\mathbf{f}\xi \ge f_0$ *is a VI, then so is* $\mathbf{f}^+\xi \ge f_0$ *where* $\mathbf{f}^+ = \max\{\mathbf{0}, \mathbf{f}\}$.

Example 2. From (1) we get that the inequality

$$-x_2 - y_1 \le -2 \tag{4}$$

is a valid inequality for the mixed-integer solutions to (1) and (2).

Rule III. "Strengthening" a VI by Euclidean reduction gives a VI:

If $\mathbf{f}\xi = \sum_{j=1}^n f_j x_j \le f_0$ *is a VI with* $f_j \in \mathbb{Z}^n$ *for* $j = 1, \dots, n$ *and greatest common divisor* $\alpha = g.c.d(f_1, \dots, f_n)$, *then* $\sum_{j=1}^n (f_j/\alpha)x_j \le \lfloor f_0/\alpha \rfloor$ *is a VI, where* $\lfloor f_0/\alpha \rfloor$ *is the biggest integer less than or equal to* f_0/α.

Example 3. From (3) we get $\alpha = 6$ and hence that

$$x_1 + 2x_3 \le \lfloor 5/6 \rfloor = 0$$

and consequently, $x_1 = x_3 = 0$ in every feasible solution to (1) and (2).

Rule IV. "Strengthening" a VI by rotation gives a VI:

If $\mathbf{f}\xi = \sum_{j=1}^n f_j x_j \ge f_0$ *with* $f_j \ge 0$ *for* $0 \le j \le n$ *is a VI, then so is* $\sum_{j=1}^n \min\{f_0, f_j\} x_j \ge f_0$.

Example 4. The inequality $5x_1 + 7x_2 + x_3 \ge 3$ in nonnegative integer variables can be replaced by the stronger inequality $3x_1 + 3x_2 + x_3 \ge 3$ (see the figure). Consider next the inequality

$$4x_1 + 8x_2 + 3x_3 - 500x_4 \le 12 \tag{5}$$

in zero-one variables $x_i \in \{0, 1\}$ for $1 \le i \le 4$. Complementing $x_i' = 1 - x_i$ for $1 \le i \le 3$ we get $4x_1' + 8x_2' + 3x_3' + 500x_4 \ge 3$ which by Rule **IV** can be replaced by

$$3x_1' + 3x_2' + 3x_3' + 3x_4 \ge 3.$$

We simplify using Rule **III** and thus by reversing the complementation we get that

$$x_1 + x_2 + x_3 - x_4 \le 2$$

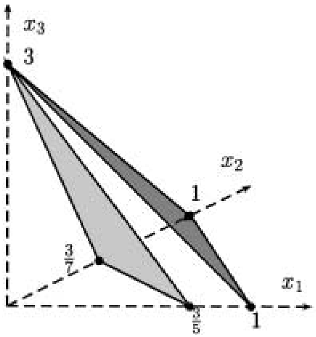

Fig. 2. Strengthening by rotation.

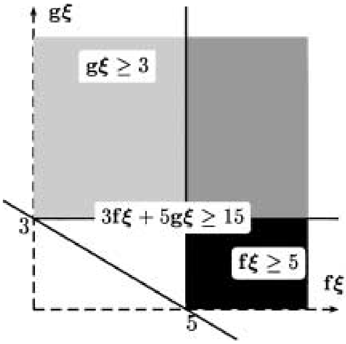

Fig. 3. Using disjunctions.

is a valid inequality for all zero-one solutions to (5) and (5) becomes redundant.

For $\mathbf{f} \in \mathbb{R}^{n+p}$ we denote by supp($\mathbf{f}$) the support of $\mathbf{f}$, i.e.,

$$\text{supp}(\mathbf{f}) = \{\ell \in \{1, \ldots, n+p\} : f_\ell \neq 0\}.$$

Rule V. "Valid disjunctions" give VIs:

a.) *Let* $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{n+p}$ *be such that*

$$\text{supp}(\mathbf{f}) \cap \text{supp}(\mathbf{g}) = \emptyset, \quad \mathbf{f}\xi \geq 0 \textit{ and } \mathbf{g}\xi \geq 0$$

for all feasible $\xi \in \mathbb{R}^{n+p}$. *If the disjunction "either* $\mathbf{f}\xi \geq a$ *or* $\mathbf{g}\xi \geq b$ *for all feasible* ξ*" is valid for some* $a \geq 0$ *and* $b \geq 0$*, then* $(b\mathbf{f} + a\mathbf{g})\xi \geq ab$ *is a VI.*

b.) *Let* $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{n+p}$ *be such that* $f_j \geq 0$ *and* $g_j \geq 0$ *for all* j *and* $H = \{j \in \{1, \ldots, n+p\} : f_j + g_j > 0\} \neq \emptyset$. *If the disjunction "either* $\mathbf{f}\xi \geq a$ *or* $\mathbf{g}\xi \geq b$ *for all feasible* ξ*" is true for some* $a > 0$ *and* $b > 0$*, then* $\sum_{j \in H} \left(\min\left\{ \frac{a}{f_j}, \frac{b}{g_j} \right\} \right)^{-1} \xi_j \geq 1$ *is a VI.*

c.) *Let* $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{n+p}$. *If the disjunction "either* $\mathbf{f}\xi \leq f_0$ *or* $\mathbf{g}\xi \leq g_0$ *for all feasible* ξ*" is true, then* $\sum_{j=1}^{n+p} \min\{f_j, g_j\}\xi_j \leq \max\{f_0, g_0\}$ *is a VI.*

Example 5. Given the disjunction $4x_1 + 5y_1 \geq 3$ or $2x_2 + 4y_2 \geq 5$ both Rule **V**a.) and **V**b.) apply and yield the valid inequality

$$\tfrac{4}{3}x_1 + \tfrac{2}{5}x_2 + \tfrac{5}{3}y_1 + \tfrac{4}{5}y_2 \geq 1.$$

Rule **V**c.) – which applies generally – gives the valid inequality

$$\tfrac{4}{3}x_1 + \tfrac{2}{3}x_2 + \tfrac{5}{3}y_1 + \tfrac{4}{3}y_2 \geq 1,$$

which is weaker than the one obtained by the application of Rules **V**a.) and **V**b.). Given the disjunction $4x_1 + 2x_2 + 5y_1 \geq 3$ or $2x_2 + 4y_2 \geq 5$ Rule **V**a.) does not apply, Rule **V**b.) gives the valid inequality $\frac{4}{3}x_1 + \frac{2}{3}x_2 + \frac{5}{3}y_1 + \frac{4}{5}y_2 \geq 1$ and the application of Rule **V**c.) gives the same weaker valid inequality as before. In the presence of negative coefficients and support overlap in the disjunction only Rule **V**c.) applies.

2 Classical valid inequalities for MIP

For $\boldsymbol{\mu} \in \mathbb{R}^m$ let $\boldsymbol{\mu}^+ = \max\{\mathbf{0}, \boldsymbol{\mu}\} = (\mu_i^+)_{i=1,\dots,m}$, where $\mu_i^+ = \max\{\mu_i, 0\}$, $\boldsymbol{\mu}^- = \min\{\mathbf{0}, \boldsymbol{\mu}\}$,

$$\lfloor \boldsymbol{\mu}\mathbf{A} \rfloor = (\lfloor \boldsymbol{\mu}\mathbf{a}_j \rfloor)_{j\in N}, \mathbf{f}^\mu = (f_1^\mu, \dots, f_n^\mu) = \boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor \quad \text{and}$$

$$f_0^\mu = \boldsymbol{\mu}\mathbf{b} - \lfloor \boldsymbol{\mu}\mathbf{b} \rfloor,$$

where $N = \{1, \dots, n\}$ and $\mathbf{a}_j$ is column j of $\mathbf{A}$. Note that $0 \leq f_j^\mu < 1$ for all $0 \leq j \leq n$. The derivation of the following valid inequality (FC) for (MIP) follows, besides Gomory's original reasoning [28], the spirit of proof arguments due to Fleischmann [19].

Proposition 1. *For every* $\boldsymbol{\mu} \in \mathbb{R}^m$ *and* $S \subseteq N$ *the following inequality is a VI for (MIP):*

$$((1 - f_0^\mu)\lfloor \boldsymbol{\mu}\mathbf{A} \rfloor - \boldsymbol{\mu}^-\mathbf{A})\mathbf{x} + \sum_{j\in S} (f_j^\mu - f_0^\mu)x_j + \min\{\boldsymbol{\mu}^+\mathbf{D}, -\boldsymbol{\mu}^-\mathbf{D}\}\mathbf{y}$$

$$\leq (1 - f_0^\mu)\lfloor \boldsymbol{\mu}\mathbf{b} \rfloor - \boldsymbol{\mu}^-\mathbf{b}. \tag{FC}$$

Proof. Introducing slack variables $\mathbf{s} \in \mathbb{R}^m$, $\mathbf{s} \geq \mathbf{0}$ it follows that

$$\boldsymbol{\mu}\mathbf{A}\mathbf{x} + \boldsymbol{\mu}\mathbf{D}\mathbf{y} + \boldsymbol{\mu}\mathbf{s} = \boldsymbol{\mu}\mathbf{b}$$

for all $(\mathbf{x}, \mathbf{y}) \in DM$ and all $\boldsymbol{\mu} \in \mathbb{R}^m$. Consequently,

$$(\boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor)\mathbf{x} - \sum_{j \in S} x_j + \boldsymbol{\mu}\mathbf{D}\mathbf{y} + \boldsymbol{\mu}\mathbf{s} - f_0^{\mu}$$
$$= \lfloor \boldsymbol{\mu}\mathbf{b} \rfloor - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor \mathbf{x} - \sum_{j \in S} x_j \in \mathbb{Z} \tag{6}$$

for all $(\mathbf{x}, \mathbf{y}) \in DM$. Since every integer number is either greater than or equal to 0 or less than or equal to -1 we get the valid disjunction

$$\text{either} \quad (\boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor)\mathbf{x} - \sum_{j \in S} x_j + \boldsymbol{\mu}\mathbf{D}\mathbf{y} + \boldsymbol{\mu}\mathbf{s} - f_0^{\mu} \geq 0 \tag{i}$$

$$\text{or} \quad (\boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor)\mathbf{x} - \sum_{j \in S} x_j + \boldsymbol{\mu}\mathbf{D}\mathbf{y} + \boldsymbol{\mu}\mathbf{s} - f_0^{\mu} \leq -1 \tag{ii}$$

for all $(\mathbf{x}, \mathbf{y}) \in DM$. Consequently from Rule **II** by weakening (i) and (ii) it follows that

$$\text{either} \quad \sum_{j \in N-S} f_j^{\mu} x_j + (\boldsymbol{\mu}\mathbf{D})^+ \mathbf{y} + \boldsymbol{\mu}^+ \mathbf{s} \geq f_0^{\mu} \tag{i*}$$

$$\text{or} \quad -\sum_{j \in S} (1 - f_j^{\mu}) x_j + (\boldsymbol{\mu}\mathbf{D})^- \mathbf{y} + \boldsymbol{\mu}^- \mathbf{s} \leq -(1 - f_0^{\mu}) \tag{ii*}$$

for all $(\mathbf{x}, \mathbf{y}) \in DM$. Since $1 - f_0^{\mu} > 0$ we can rewrite (ii*) as

$$\frac{1}{1 - f_0^{\mu}} \left(\sum_{j \in S} (1 - f_j^{\mu}) x_j - (\boldsymbol{\mu}\mathbf{D})^- \mathbf{y} - \boldsymbol{\mu}^- \mathbf{s} \right) \geq 1. \tag{ii**}$$

It follows that Rule **V**a.) applies to the valid disjunction (i*) and (ii**) and hence

$$\sum_{j \in N-S} f_j^{\mu} x_j + (\boldsymbol{\mu}\mathbf{D})^+ \mathbf{y} + \boldsymbol{\mu}^+ \mathbf{s} + \frac{f_0^{\mu}}{1 - f_0^{\mu}} \left(\sum_{j \in S} (1 - f_j^{\mu}) x_j - (\boldsymbol{\mu}\mathbf{D})^- \mathbf{y} - \boldsymbol{\mu}^- \mathbf{s} \right)$$
$$\geq f_0^{\mu} \tag{7}$$

is a VI for (MIP). To eliminate $\boldsymbol{\mu}^+\mathbf{s}$ and $\boldsymbol{\mu}^-\mathbf{s}$ from (7) we use the identities

$$\boldsymbol{\mu}^+ \mathbf{s} = \boldsymbol{\mu}^+ \mathbf{b} - \boldsymbol{\mu}^+ \mathbf{A}\mathbf{x} - \boldsymbol{\mu}^+ \mathbf{D}\mathbf{y} \quad \text{and} \quad \boldsymbol{\mu}^- \mathbf{s} = \boldsymbol{\mu}^- \mathbf{b} - \boldsymbol{\mu}^- \mathbf{A}\mathbf{x} - \boldsymbol{\mu}^- \mathbf{D}\mathbf{y}.$$

Since e.g. $(\boldsymbol{\mu}\mathbf{A})^+ + (\boldsymbol{\mu}\mathbf{A})^- = \boldsymbol{\mu}\mathbf{A} = \boldsymbol{\mu}^+\mathbf{A} + \boldsymbol{\mu}^-\mathbf{A}$, all $(\mathbf{x}, \mathbf{y}) \in DM$ satisfy the inequality

$$-\lfloor \boldsymbol{\mu}\mathbf{A} \rfloor \mathbf{x} + \frac{1}{1 - f_0^{\mu}} \boldsymbol{\mu}^- \mathbf{A}\mathbf{x} + \sum_{j \in S} \frac{(f_0^{\mu} - f_j^{\mu})}{1 - f_0^{\mu}} x_j - \frac{1}{1 - f_0^{\mu}} ((\boldsymbol{\mu}\mathbf{D})^- - \boldsymbol{\mu}^- \mathbf{D})\mathbf{y}$$
$$\geq \frac{1}{1 - f_0^{\mu}} \boldsymbol{\mu}^- \mathbf{b} - \lfloor \boldsymbol{\mu}\mathbf{b} \rfloor.$$

From $(\boldsymbol{\mu}\mathbf{D})^{-} - \boldsymbol{\mu}^{-}\mathbf{D} = \min\{\boldsymbol{\mu}^{+}\mathbf{D}, -\boldsymbol{\mu}^{-}\mathbf{D}\}$ it follows that (FC) is valid for (MIP). ■

It is clear that an enormous variety of valid inequalities for (MIP) results from the application of Proposition 1. For instance, choosing $\boldsymbol{\mu} = \mathbf{u}^i \in \mathbb{R}^m$, i.e., the *i-th* unit vector, and

$$S = \{j \in N : f_j^i > f_0^i\},$$

we get from (FC) the recent *mixed-integer rounding* (MIR) *inequality*, see [49], as a special case, i.e., for $i \in \{1, \ldots, m\}$ the inequality

$$\sum_{j \in N} \left(\lfloor a_j^i \rfloor + \frac{(f_j^i - f_0^i)^+}{1 - f_0^i} \right) x_j + \frac{1}{1 - f_0^i} \min\{\mathbf{d}^i, \mathbf{0}\}\mathbf{y} \leq \lfloor b_i \rfloor \tag{MIR}$$

is valid for (MIP), where $a_j^i = \mathbf{u}^i \mathbf{a}_j$, $f_j^i = f_j^{\mathbf{u}^i}$, $\mathbf{d}^i = \mathbf{u}^i \mathbf{D}$ and $b_i = \mathbf{u}^i \mathbf{b}$. The question is how to find a "cut", i.e., a valid inequality that cuts off e.g. the optimum solution to the linear programming relaxation (MIP$_{LP}$). To answer the question let us first assume that all slack variables of the constraint set of (MIP) have been included into the flow variables of the problem and study the problem in equality form:

$$\max\{\mathbf{cx} + \mathbf{dy} : \mathbf{Ax} + \mathbf{Dy} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \text{ and integer}, \mathbf{y} \geq \mathbf{0}\}, \tag{MIP$^=$}$$

where we assume again that $\mathbf{D}$ is a $m \times p$ matrix and that all data are rational numbers.

Proposition 2. *For every $\boldsymbol{\mu} \in \mathbb{R}^m$ and $S \subseteq N$ the following inequality is a VI for (MIP$^=$):*

$$\sum_{j \in N-S} f_j^{\mu} x_j + \sum_{j \in S} \frac{f_0^{\mu}(1 - f_j^{\mu})}{1 - f_0^{\mu}} x_j + \max\left\{\boldsymbol{\mu}\mathbf{D}, \frac{-f_0^{\mu}}{1 - f_0^{\mu}} \boldsymbol{\mu}\mathbf{D}\right\}\mathbf{y} \geq f_0^{\mu}. \tag{FC$^=$}$$

Moreover, a best possible choice for S for fixed $\boldsymbol{\mu} \in \mathbb{R}^m$ with $f_0^{\mu} > 0$ gives the VI

$$\sum_{j \in N} \min\left\{\frac{f_j^{\mu}}{f_0^{\mu}}, \frac{1 - f_j^{\mu}}{1 - f_0^{\mu}}\right\} x_j + \max\left\{\frac{1}{f_0^{\mu}} \boldsymbol{\mu}\mathbf{D}, \frac{-1}{1 - f_0^{\mu}} \boldsymbol{\mu}\mathbf{D}\right\}\mathbf{y} \geq 1. \tag{FC$^\#$}$$

Proof. We argue for arbitrary $S \subseteq N$ like in the proof of Proposition 1. Here (7) becomes

$$\sum_{j \in N-S} f_j^{\mu} x_j + (\boldsymbol{\mu}\mathbf{D})^{+}\mathbf{y} + \frac{f_0^{\mu}}{1 - f_0^{\mu}} \left(\sum_{j \in S} (1 - f_j^{\mu}) x_j - (\boldsymbol{\mu}\mathbf{D})^{-}\mathbf{y} \right) \geq f_0^{\mu}.$$

Simplifying essentially as before shows the validity of (FC$^=$) for (MIP$^=$). Since all coefficients of the left-hand side of inequality (FP$^=$) are nonnegative, we choose for fixed $\boldsymbol{\mu} \in \mathbb{R}^m$ with $f_0^{\mu} > 0$ the set $S \subseteq N$ such that the intersection with the coordinate axis of x_j is as large as possible, i.e., we select $S = \{j \in N : f_j^\mu > f_0^\mu\}$ to obtain (FC$^\#$). ■

2.1 Gomory's mixed-integer cut

Let now $\mathbf{B}$ be any basis for the linear programming relaxation (MIP$_{LP}^=$) of (MIP$^=$), e.g. an optimal basis found by the simplex algorithm when applied to solve

$$\max\{\mathbf{cx} + \mathbf{dy} : \mathbf{Ax} + \mathbf{Dy} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}\}. \tag{MIP$_{LP}^=$}$$

Denote by $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ the transformed right-hand side and let $r \in \{1, \ldots, m\}$ be such that the associated basic variable $k(r)$ is an integer variable, but $\bar{b}_r \notin \mathbb{Z}$. If no such r exists and $\mathbf{B}$ is an optimal basis, then the solution to (MIP$_{LP}^=$) solves (MIP$^=$) and we are done. So suppose such $r \in \{1, \ldots, m\}$ exists and choose

$$\boldsymbol{\mu} = \alpha \mathbf{u}_r \mathbf{B}^{-1} \tag{8}$$

where $\mathbf{u}_r \in \mathbb{R}^m$ is the *r-th* unit vector and $\alpha \in \mathbb{Z}$ is any integer. We denote by

$$\bar{\mathbf{a}}_j = \mathbf{B}^{-1}\mathbf{a}_j \quad \text{and} \quad \bar{\mathbf{d}}_j = \mathbf{B}^{-1}\mathbf{d}_j$$

the transformed columns of $\mathbf{A}$ and $\mathbf{D}$, respectively. Denote for the choice (8) of $\boldsymbol{\mu}$

$$f_j(\alpha) = \alpha \bar{a}_j^r - \lfloor \alpha \bar{a}_j^r \rfloor \quad \text{and} \quad f_0(\alpha) = \alpha \bar{b}_r - \lfloor \alpha \bar{b}_r \rfloor. \tag{9}$$

For $\alpha = 1$ we have $f_0(\alpha) > 0$ by the choice of r. More generally, assume that $\alpha \in \mathbb{Z}$ is such that $f_0(\alpha) > 0$. The valid inequality (FC$^=$) becomes

$$\sum_{j \in N-S} f_j(\alpha) x_j + \sum_{j \in S} \frac{f_0(\alpha)(1 - f_j(\alpha))}{1 - f_0(\alpha)} x_j + \sum_{j : \bar{d}_j^r > 0} \alpha \bar{d}_j^r y_j$$
$$- \frac{f_0(\alpha)}{1 - f_0(\alpha)} \sum_{j : \bar{d}_j^r < 0} \alpha \bar{d}_j^r y_j \geq f_0(\alpha). \tag{GM$^=$}$$

By the choice (8) of $\boldsymbol{\mu}$ and $\alpha \in \mathbb{Z}$, $f_j(\alpha) = 0$ and $\bar{d}_j^r = 0$ for all basic integer and flow variables. Hence the left-hand side of (GM$^=$) for the optimal linear programming solution of (MIP$_{LP}^=$) equals zero. Since $f_0(\alpha) > 0$, (GM$^=$) is thus a cut for (MIP$_{LP}^=$). Evidently, the choice of $\boldsymbol{\mu}$ in (8) is only one of many possibilities to generate cuts of the general form (FC$^=$). Choosing for fixed $\alpha \in \mathbb{Z}$, $\alpha > 0$, the set $S = \{j \in N : f_j(\alpha) > f_0(\alpha)\}$, we obtain from (FC$^\#$)

$$\sum_{j=1}^{n} \min\left\{\frac{f_j(\alpha)}{f_0(\alpha)}, \frac{1-f_j(\alpha)}{1-f_0(\alpha)}\right\} x_j + \sum_{j=1}^{p} \alpha \max\left\{\frac{\bar{d}_j^r}{f_0(\alpha)}, \frac{-\bar{d}_j^r}{1-f_0(\alpha)}\right\} y_j \geq 1. \tag{GM$^{\#}$}$$

This is Gomory's classical "mixed-integer cut" [28], which is "basis-dependent"; see (8).

In terms of the two questions raised in the introduction it follows that the answer to the first one is positive. The answer to the second one is, however, negative if we use the cut (GM$^{\#}$). The following is a simplification of the original example by White [81].

Example 5. (*Infinite convergence of (GM$^{\#}$) cuts* [59], [74], [81]). Consider the program (MIP)

$$\begin{array}{lrrrcl}
\max & & & y_1 & & \\
\textit{such that} & x_1 & +x_2 & +y_1 & \leq & 2 \\
 & -x_1 & & +y_1 & \leq & 0 \\
 & & -x_2 & +y_1 & \leq & 0 \\
 & x_1 \geq 0 & x_2 \geq 0 & y_1 \geq 0 & &
\end{array}$$

where x_1 and x_2 are the integer variables. We shall use the algorithm A of Figure 1 with the following cut generation mechanism: we select $\alpha = 1$ and at every iteration we derive the cuts (GM$^{\#}$) for the two integer variables x_1 and x_2 whenever possible. We then express the cuts in terms of the original variables and drop constraints that become redundant in the process. We claim that after $\ell \geq 0$ iterations the current linear program (LP$_\ell$) is

$$\begin{array}{lrrrcl}
\max & & & y_1 & & \\
\textit{such that} & x_1 & +x_2 & +y_1 & \leq & 2 \\
 & -x_1 & & +(\ell+1)y_1 & \leq & 0 \\
 & & -x_2 & +(\ell+1)y_1 & \leq & 0 \\
 & x_1 \geq 0 & x_2 \geq 0 & y_1 \geq 0 & &
\end{array}$$

and that the optimal solution $(\mathbf{x}^\ell, \mathbf{y}^\ell)$ to (LP$_\ell$) is given by

$$x_1^\ell = x_2^\ell = \frac{2\ell+2}{2\ell+3}, \quad y_1^\ell = \frac{2}{2\ell+3}.$$

To prove it, we consider the basis and its inverse

$$\mathbf{B} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & \ell+1 \\ 0 & -1 & \ell+1 \end{pmatrix}, \quad \mathbf{B}^{-1} = \frac{1}{2\ell+3}\begin{pmatrix} \ell+1 & -(\ell+2) & \ell+1 \\ \ell+1 & \ell+1 & -(\ell+2) \\ 1 & 1 & 1 \end{pmatrix}.$$

Calculating the dual solution $\mathbf{v}^\ell = \mathbf{c_B}\mathbf{B}^{-1} = \frac{1}{2\ell+3}(1 \ 1 \ 1)$ it follows that the above solution $(\mathbf{x}^\ell, \mathbf{y}^\ell)$ is indeed an optimal solution to (LP$_\ell$) for all $\ell \geq 0$. Consequently, from (8) we get $\boldsymbol{\mu}^1 = \mathbf{u}_1\mathbf{B}^{-1} = \frac{1}{2\ell+3}(\ell+1 \ -(\ell+2) \ \ell+1)$

and thus from ($GM^=$) or ($GM^{\#}$) the cut

$$\frac{\ell+1}{2\ell+3}s_1 - \frac{(2\ell+2)/(2\ell+3)}{1-(2\ell+2)/(2\ell+3)}\left(-\frac{\ell+2}{2\ell+3}\right)s_2 + \frac{\ell+1}{2\ell+3}s_3 \geq \frac{2\ell+2}{2\ell+3}.$$

Likewise with $\boldsymbol{\mu}^2 = \mathbf{u}_2\mathbf{B}^{-1} = \dfrac{1}{2\ell+3}(\ell+1 \;\; \ell+1 \;\; -(\ell+2))$ we get the cut

$$\frac{\ell+1}{2\ell+3}s_1 + \frac{\ell+1}{2\ell+3}s_2 - \frac{(2\ell+2)/(2\ell+3)}{1-(2\ell+2)/(2\ell+3)}\left(-\frac{\ell+2}{2\ell+3}\right)s_3 \geq \frac{2\ell+2}{2\ell+3}$$

where in both cases

$$s_1 = 2 - x_1 - x_2 - y_1 \geq 0, \quad s_2 = x_1 - (\ell+1)y_1 \geq 0,$$

$$s_3 = x_2 - (\ell+1)y_1 \geq 0$$

are the respective slack variables of (LP_ℓ), which are flow variables for the corresponding ($MIP^=$). Simplifying and clearing common divisors we get the two cuts

$$-x_1 + (\ell+2)y_1 \leq 0, \quad -x_2 + (\ell+2)y_1 \leq 0.$$

But $-x_1 + (\ell+2)y_1 \leq 0$ and $-y_1 \leq 0$ imply that$-x_1 + (\ell+1)y_1 \leq 0$ and thus the inequality $-x_1 + (\ell+1)y_1 \leq 0$ becomes redundant once we add the new cut. Likewise the inequality $-x_2 + (\ell+1)y_1 \leq 0$ becomes redundant. After $\ell \geq 0$ iterations the application of the above cut generation mechanism thus produces (LP_ℓ) and

$$\lim_{\ell\to\infty} y_1^\ell = \lim_{\ell\to\infty} \frac{2}{2\ell+3} = 0$$

shows the infinite convergence of Algorithm A. ■

In the example the linear programming relaxation is a 3-dimensional polytope, whereas the convex hull of the discrete-mixed set DM is 2-dimensional because $y_1 = 0$ for all feasible mixed-integer solutions in this case. This may explain why infinite convergence is obtained. Under certain technical assumptions, however, convergence of Algorithm A has been established.

Theorem 1 (Gomory [28]): *The cutting plane algorithm for ($MIP^=$) with the cut ($GM^=$) for $\alpha = 1$ converges in a finite number of steps if*

(i) *one chooses a single integer variable x_j for cut generation by a "least index" rule, i.e., the first integer that is non-integer valued in the current linear programming solution.*
(ii) *one requires that the optimum objective function value of ($MIP^=$) is an integer number, i.e., $x_0 - \mathbf{cx} - \mathbf{dy} = 0$ is used in the cut generation with the requirement that x_0 is an integer variable.*
(iii) *one uses the "lexicographic" version of the simplex algorithm.*

Of the technical assumptions of the theorem (ii) is the most restrictive one. In the above example this assumption is tantamount to requiring that all variables are integers.

2.2 *Gomory's fractional cut*

We consider now the case of a pure integer program with integer data separately, i.e., we consider the problem in equality form

$$\max\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \text{ and integer}\}, \qquad (\mathrm{IP}^{=})$$

where we assume that $\mathbf{A}$ is a $m \times n$ matrix of integers and that the right-hand side $\mathbf{b}$ is a vector of m integer numbers.

Proposition 3. *For every $\boldsymbol{\mu} \in \mathbb{R}^m$ the inequality*

$$(\boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor)\mathbf{x} \geq \boldsymbol{\mu}\mathbf{b} - \lfloor \boldsymbol{\mu}\mathbf{b} \rfloor \qquad (\mathrm{GP}^{=})$$

is valid for ($IP^{=}$). Moreover, the surplus variable in ($GP^{=}$) is an integer variable.

Proof. We proceed like in the proof of Proposition 1. Equation (6) becomes for $S = \varnothing$

$$(\boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor)\mathbf{x} - f_0^{\mu} = \lfloor \boldsymbol{\mu}\mathbf{b} \rfloor - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor \mathbf{x} \in \mathbb{Z} \qquad (10)$$

for all $\mathbf{x} \in \mathbb{Z}^n$. But $\boldsymbol{\mu}\mathbf{A} \geq \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor$, $\mathbf{x} \geq \mathbf{0}$ and $0 \leq f_0^{\mu} < 1$. Consequently, since the right-hand side of (10) must be integer and the left-hand side of (10) is greater than -1 for all $\mathbf{x} \geq \mathbf{0}$ it follows that

$$(\boldsymbol{\mu}\mathbf{A} - \lfloor \boldsymbol{\mu}\mathbf{A} \rfloor)\mathbf{x} - f_0^{\mu} = x_{n+1} \geq 0$$

for all $\mathbf{x} \in \mathbb{Z}^n$, $\mathbf{x} \geq \mathbf{0}$ and the surplus variable x_{n+1} is a nonnegative integer. ■

If we add the valid inequality ($\mathrm{GP}^{=}$) in equation form to ($\mathrm{IP}^{=}$), then the augmented problem is again of the general form ($\mathrm{IP}^{=}$) which permits the iterative application of the basic idea. Let $\mathbf{B}$ be any basis of the associated relaxed linear program

$$\max\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \qquad (\mathrm{IP}^{=}_{LP})$$

e.g. an optimal basis for ($\mathrm{IP}^{=}_{LP}$). Like in the case of (8) we set $\boldsymbol{\mu} = \alpha \mathbf{u}_r \mathbf{B}^{-1}$ where $\alpha \in \mathbb{Z}$ and choose $r \in \{1, \ldots, m\}$ such that $\bar{b}_r \notin \mathbb{Z}$ where $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$. Then using the same notation as in (9) the valid inequality ($\mathrm{GP}^{=}$) becomes

$$\sum_{j \in N} f_j(\alpha) x_j \geq f_0(\alpha). \qquad (\mathrm{GP}^{=})$$

Like in the case of ($GM^=$) $f_j(\alpha) = 0$ for all basic variables. Thus ($GP^=$) is a cut because we can assume that $f_0(\alpha) > 0$. This is Gomory's classical "fractional cut" [26], which like ($GM^=$) and ($GM^\#$) is basis-dependent.

Theorem 2 (Gomory [26]). *Under mild technical assumptions about the cut generation mechanism and for integer data* $(\mathbf{A}, \mathbf{b})$ *the cutting plane algorithm converges in finitely many steps when ($GP^=$) with* $\alpha = 1$ *is used as a cut.*

Example 6. (*Speed of convergence of ($GP^=$) cuts* [59]). Consider the integer program (IP)

$$\begin{array}{llcccc} & \max & x_1 & +x_2 & & \\ \textit{such that} & & -Kx_1 & +x_2 & \leq & 1 \\ & & Kx_1 & +x_2 & \leq & K+1 \\ & & x_1 \geq 0 & x_2 \geq 0 & & \end{array}$$

where both x_1 and x_2 are integer variables and $K \geq 2$ is any integer number. Let us agree to select $\alpha = 1$ and to generate cuts by a least index rule, i.e., we will choose variable x_1 as long as it is possible. Note that the slacks in the two constraints of (IP) are integer because of the integrality of the data. Thus adding slack variables we bring (IP) into the form ($IP^=$). As we did in the previous example we express the cut in the original variables x_1 and x_2, clear common divisors and drop redundant constraints. We leave to the reader to show that after $\ell \geq 0$ applications of ($GP^=$) we get the linear program (LP_ℓ)

$$\begin{array}{llcccc} & \max & x_1 & +x_2 & & \\ \textit{such that} & & -(K-\ell)x_1 & +x_2 & \leq & 1 \\ & & Kx_1 & +x_2 & \leq & K+1 \\ & & x_1 \geq 0 & x_2 \geq 0 & & \end{array}$$

the optimal solution $\mathbf{x}^\ell$ to which is given by

$$x_1^\ell = \frac{K}{2K-\ell}, \quad x_2^\ell = 1 + \frac{K(K-\ell)}{2K-\ell}.$$

Thus K cuts ($GP^=$) must be applied to find the optimal integer solution $(x_1, x_2) = (1, 1)$. Now choose e.g. $K = 10^{10}$ and draw your own conclusions. By contrast, in this example the application of ($GM^\#$) to (LP_0), in spite of its potential for infinite convergence, gives the cut $x_2 \leq 1$ which suffices to obtain the optimal integer solution. ■

The cut ($GM^\#$) for $p = 0$ is, in general, "stronger" than ($GM^=$) and ($GP^=$), but the corresponding surplus variable is, *a priori*, a flow variable.

2.3 Summary

Work on cutting planes begins with [8], [18], [26], [27], [28] and [51]. The 1950's, 1960's and early 1970's were characterized by attempts to solve pure and mixed integer programming problems using valid inequalities and "cuts" of the variety as outlined here and other ones as well, c.f. [3], [39], [40], [87]

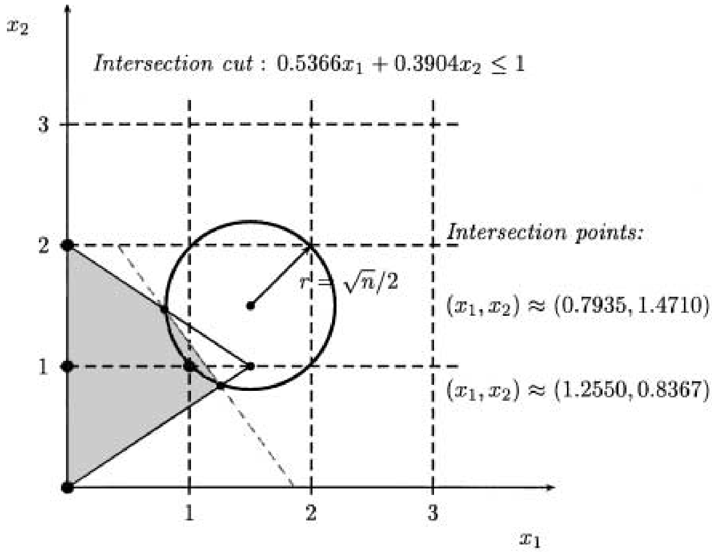

Fig. 4. Intersection cuts from the sphere.

and [22], [57], [74], [77], [85], [88] for more. (The "intersection cuts" of [3], [87] are illustrated in Figure 4.) Suffice it to say that – to the best of the author's knowledge – no commercial problem solver has implemented the pertinent research work in any detail until the mid 1990's. Indeed, general "cutting planes" for integer programming were largely forgotten – until the recent revival of ($GM^{\#}$), see below. The reason for this – in our humble view – is twofold: first, a theoretically exact implementation of the cutting plane theory reviewed here requires LP solvers that work in **exact arithmetic**, i.e., that carry out their calculations over the field of rational numbers. We shall come back to this issue below. Secondly, there were plenty of early attempts to implement cutting plane theory numerically nevertheless, c.f. [19], [20], [35], [42], [43], [54], [56], [76], [78], [80] and probably more. Most of the pertaining computational studies used the fractional cut ($GP^{=}$) or its "all-integer" variant [25], [27], [29], [82], [86], whereas the stronger mixed-integer cut ($GM^{\#}$) was studied to a lesser degree; probably because it makes a "mixed" integer program even more "mixed" as the surplus variable in ($GM^{\#}$) is in general a flow variable. The derivation of ($GM^{\#}$) and ($GP^{=}$), respectively, uses substantially different reasoning, see (7) and (10).

None of the studies reported in the early literature on the subject showed significant numerical success, indeed some reported outright disastrous results, c.f. [42], [43], [76] and Chapter 13 (written by R. Woolsey) in [74]. The only known exception to the rule was G. Martin's work [54] on set covering and traveling salesman problems of small to medium size; see also [56] where a discussion of implementational difficulties of Gomory's fractional cuts can be found. Luckily, branch-and-bound [18], [44] and implicit enumeration [2], [23], [75] sufficed to solve most problems of its time, see e.g. the surveys [6], [9], [24]. Rightly or wrongly – classical cutting plane theory fell into oblivion by the end of the 1970's, see e.g. [21], [45] and also the extensive treatment of cutting planes in e.g. [22] or [74] as opposed to their treatment in more recent texts such as [57] and [85].

3 Branch-and-cut

The early 1970's were the start of a mathematically different approach to mixed-integer programming, one that continues to be pursued actively to date. The polyhedrality of the convex hull of the discrete-mixed set DM of *bounded* (mixed-) integer programs being a trivial consequence of boundedness, the *facial structure* of many *problem-specific* polytopes, such as those associated with set packing, traveling salesman, knapsack, simple plant location, linear ordering, acyclic subgraph, quadratic zero-one and many other combinatorial optimization problems, was studied in depth, c.f. [4], [13], [14], [31], [33], [58], [60], [61], [62], [63], [64], [83] and many, many more. These studies were extended to the mixed-integer case already in the early 1980's, c.f. [46], [47], [48], [71], [72], [73], [84] and more. As a result, partial linear descriptions by way of facet-defining inequalities of many pure and mixed integer programming problems are known to date. The facet-defining inequalities of the associated polytopes and polyhedra are strongest possible cutting-planes for the respective problems. The basic hypothesis to be tested was thus: do such cuts "work" in numerical practice or not? A useful corollary to the outcome of such experimentation is, of course, that

> *if strongest possible cuts for* (MIP) *do not work, then the whole cutting plane approach to* (MIP) *can safely be put to rest.*

As the computational record has shown, the use of facet-defining inequalities and/or their various approximations as cuts has pushed the limits of numerical computability in mixed-integer programming far beyond those that are attainable by branch-and-bound, c.f. [1], [16], [17], [30], [31], [34], [36], [37], [38], [41], [48], [67], [68], [69], [79] and not only many more, but the pertaining literature seems to keep growing year by year. See [50] for an excellent, recent survey.

The trouble with this approach – which became evident in the 1970's as well – is that, perhaps because of the NP-hard characteristic of all of the problems mentioned above, it is unlikely that complete (minimal and ideal) linear descriptions of the associated polytopes and polyhedra can be obtained. Indeed, the "pure" cutting-plane approach of Figure 1 using as cuts *only* facet-defining inequalities (rather than arbitrary valid inequalities) assumes that such a description is available and that it is algorithmically "identifiable", i.e., that we know how to solve the corresponding "separation" problem as well. (One of the theoretically most interesting by-products of this line of research is the polynomial-time equivalence of optimization and separation, see e.g. [32] and [65], Remark 9.20 and point 10.2(f), for a precise statement and proof of this fundamental theorem.) Perhaps as a result of the NP-hard characteristic, the pure approach has been successful (so far) for only a few classes of integer programs, c.f. [30], [66], [70] and perhaps some others as well.

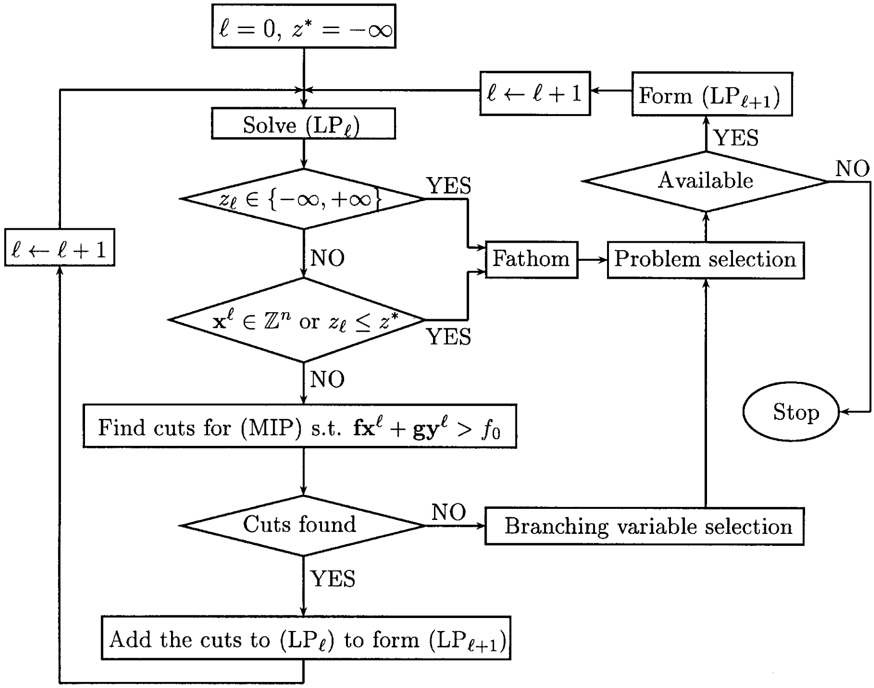

Fig. 5. Branch-and-Cut algorithm B for MIP.

The way out of this dilemma was to combine cutting planes based on facet-defining inequalities with the oldest and simplest approach to integer and mixed-integer programming problems, namely branch-and-bound. Branch-and-bound (see Figures 5 and 6) tries to locate the (mixed-) integer optimum by successively splitting the "current" problem into two or more smaller subproblems while organizing the search for a global optimum by way of a (binary) search tree that keeps track of all (active) subproblems that are generated. To be able to "marry" such an enumerative approach with cutting planes efficiently one needs cuts that are valid across the entire search tree, i.e., cuts that are valid "globally" across all subproblems that may be generated during the search for a global optimum. Facet-defining inequalities of the overall polytope or polyhedron have this property automatically and are thus ideally suited to be adapted to search tree methods. Indeed, the numerical success on the academic research side of things that resulted from the actual application of this basic idea has been such that most of the producers of (serious) commercial software for the solution of (MIP) like CPLEX, LINGO, OSL and XPRESS-MP have incorporated – since the late 1990's – some (or most) of the elements of branch-and-cut into their products; see [10], [15].

In Figure 5 we give the outline of a "branch-and-cut" (B&C) algorithm for (MIP) that combines the "pure" cutting-plane algorithm of Figure 1 with branch-and-bound. The left half of the flow-chart is more or less like Figure 1 except that the possibility that no cuts are found (or are discarded for some reason or another) is allowed for explicitly. In the latter case we proceed like in branch-and-bound: using some clever heuristic for selecting a basic integer variable x_j with noninteger value $\bar{b}_r$, say, we create two new problems by requiring that $x_j \leq \lfloor \bar{b}_r \rfloor$ on the "down" branch and that $x_j \geq \lceil \bar{b}_r \rceil$ on the "up" branch. The corresponding two new problems are put on a "problem stack". The algorithm proceeds by selecting – again by some clever heuristic selection mechanism – some problem from the stack for further processing. If no more problem is "available" the problem stack is empty and the original problem (MIP) is solved. z^* in the Figure 5 is initially set to $-\infty$ and updated in the box "Fathom" whenever a (better) feasible (mixed-) integer solution has been obtained, i.e., the deletion of subproblems on the stack that do not need to be processed any further is done like in basic branch-and-bound.

The traditional school of thought in (mixed-) integer programming, i.e., the one that tried to get away with arbitrary "cuts" and valid inequalities in the solution of (MIP), having lost steam long time ago, it remained to revive these classical ideas, suitably adapted, in a "branch-and-cut" framework. This was done recently by Balas et al. [5] who show that the cut ($GM^{\#}$) for mixed zero-one programs can be integrated into a branch-and-cut algorithm; see also [10], [11], [15], [49], [50].

3.1 Invalidity of classical cuts in B&C

The fact that classical cuts are basis-dependent prohibits *a priori* their use in branch-and-cut where "global" validity, i.e., validity of the cuts across all branches of the search tree, is desirable. The obvious alternative of storing the cuts that are specific to each node of the search tree ("local" cuts) separately is generally ruled out as being too costly in terms of storage (and computational) requirements. (As two referees pointed out, there is currently some experimentation with the generation of local cuts, see [15], [41], [52], but in our view its effect on truly large-scale optimization remains to be seen.)

Global invalidity of e.g. Gomory's classical cuts ($GM^{\#}$) and ($GP^{=}$) for branch-and-cut has been known for many years, i.e., the *raw* application of such cuts in branch-and-cut is, in general, mathematically incorrect. The following example from [5] illustrates this point for ($GM^{\#}$), but similar examples for ($GP^{=}$) are easily constructed as well.

Example 7. (*Global invalidity of ($GM^{\#}$) in B&C*). Consider the pure integer program

$$\begin{array}{lrrcrcrcrcr} & \min & x_0 = & 7x_1 & + & 3x_2 & + & 4x_3 & & \\ \textit{subject to} & & & x_1 & + & 2x_2 & + & 3x_3 & \geq & 8 \\ & & & 3x_1 & + & x_2 & + & x_3 & \geq & 5 \end{array}$$

where x_1, x_2 and x_3 are nonnegative integer variables. Optimizing by the simplex algorithm gives the reduced equation format:

$$\begin{array}{lllrrcrcrcr} x_0 & & & - & \frac{3}{5}x_3 & - & \frac{2}{5}x_4 & - & \frac{11}{5}x_5 & = & 14\frac{1}{5} \\ & x_1 & & - & \frac{1}{5}x_3 & + & \frac{1}{5}x_4 & - & \frac{2}{5}x_5 & = & \frac{2}{5} \\ & & x_2 & + & \frac{8}{5}x_3 & - & \frac{3}{5}x_4 & + & \frac{1}{5}x_5 & = & 3\frac{4}{5} \end{array}$$

where x_4 and x_5 are the (integer) surplus variables. Applying (GM$^{\#}$) to the second and third row, respectively, we find the cuts

$$\tfrac{1}{3}x_3 + \tfrac{1}{2}x_4 + \tfrac{2}{3}x_5 \geq 1, \quad \tfrac{3}{4}x_3 + \tfrac{1}{2}x_4 + \tfrac{1}{4}x_5 \geq 1,$$

or when expressed in the original variables (after clearing common divisors)

$$3x_1 + 2x_2 + 3x_3 \geq 10, \quad x_1 + x_2 + 2x_3 \geq 5.$$

Adding the cuts and reoptimizing using e.g. the dual simplex algorithm we find the (unique) optimal integer solution $x_1 = 0$, $x_2 = 5$, $x_3 = 0$ with an objective function value of $x_0 = 15$. Suppose now that, after having obtained the first optimal equation format, we decide to branch on variable x_2 *prior* to adding any cuts, i.e., we shall create two new problems corresponding to $x_2 \leq \lfloor 3\frac{4}{5} \rfloor = 3$ and $x_2 \geq \lceil 3\frac{4}{5} \rceil = 4$, respectively, from the original one. Letting $x_6^d = 3 - x_2 \geq 0$ be the slack variable on the "down" branch we obtain by reoptimizing the equation format

$$\begin{array}{ccccccccccc}
x_1 & & & + & \frac{1}{8}x_4 & - & \frac{3}{8}x_5 & - & \frac{1}{8}x_6^d & = & \frac{1}{2} \\
 & x_2 & & & & & & + & x_6^d & = & 3 \\
 & & x_3 & - & \frac{3}{8}x_4 & + & \frac{1}{8}x_5 & - & \frac{5}{8}x_6^d & = & \frac{1}{2}
\end{array}$$

Applying (GM$^{\#}$) to the first and third row, respectively, we find the cuts

$$\tfrac{1}{4}x_4 + \tfrac{3}{4}x_5 + \tfrac{1}{4}x_6^d \geq 1, \quad \tfrac{3}{4}x_4 + \tfrac{1}{4}x_5 + \tfrac{3}{4}x_6^d \geq 1,$$

or when expressed in the original variables (after clearing common divisors)

$$5x_1 + 2x_2 + 3x_3 \geq 12, \quad 3x_1 + 2x_2 + 5x_3 \geq 12,$$

both of which cut off the optimal solution $x_1 = x_3 = 0, x_2 = 5$. Letting $x_6^u = x_2 - 4 \geq 0$ be the surplus variable on the "up" branch we obtain by reoptimizing the equation format

$$\begin{array}{ccccccccccc}
x_1 & & + & \frac{1}{3}x_3 & & & - & \frac{1}{3}x_5 & + & \frac{1}{3}x_6^u & = & \frac{1}{3} \\
 & x_2 & & & & & & & - & x_6^u & = & 4 \\
 & & - & \frac{8}{3}x_3 & + & x_4 & - & \frac{1}{3}x_5 & - & \frac{5}{3}x_6^u & = & \frac{1}{3}
\end{array}$$

Applying (GM$^{\#}$) to the first and third row, respectively, we find the cut $x_3 + \frac{1}{2}x_5 + x_6^u \geq 1$, or when expressed in the original variables (after clearing common divisors)

$$x_1 + x_2 + x_3 \geq 5,$$

which cuts off the suboptimal, but feasible solution $x_1 = x_3 = 2, x_2 = 0$. Thus along both the down branch and the up branch the application of (GM$^{\#}$) produces cuts that are globally not valid, i.e., that are simply not valid for the original integer program. ■

3.2 Validity of post-branching cuts in B&C

To discuss the question of when classical cuts are valid in branch-and-cut more generally, we note first that all cuts that are generated *prior* to branching are by definition globally valid cuts for (MIP) or *global cuts*, for short. So suppose that branching occurs, call the first node of the search tree when branching occurs the "root" of the tree and any cut that is generated after branching has occurred a *post-branching* cut. To analyze the validity of post-branching cuts for (MIP), we assume that branching is done as discussed above by imposing upper and lower bounds on single variables only.

To keep track of the search tree we assign to every edge of the tree some unique labeling and denote by $\mathscr{L}$ the set of all (current) labels (see Figure 6). It follows that every node in the search tree is characterized by a unique path P to the root node of the form

$$x_{k(j)} \le u_j \quad \text{for all } j \in F_d, \quad x_{k(j)} \ge \ell_j \quad \text{for all } j \in F_u \tag{11}$$

where $F_d \subseteq \mathscr{L}$ and $F_u \subseteq \mathscr{L}$ are the labels of the respective "down" and "up" branches of the path $P = F_d \cup F_u$. $x_{k(j)}$ is the branching variable, $u_j \ge 0$ the upper bound for a down branch and $\ell_j > 0$ the lower bound for an up branch for any edge $j \in F_d \cup F_u$ of the path. Any integer variable x_j with $j \in N$ may occur repeatedly along the path P. For $j \in N$ we denote by

$$B_j = \text{set of branches of the path } P \text{ containing variable } x_j.$$

It follows that the linear program that is solved at the current node of the search tree is

$$\max\{\mathbf{cx} + \mathbf{dy} : \mathbf{Ax} + \mathbf{Dy} = \mathbf{b}, x_{k(j)} \le u_j \text{ for } j \in F_d, x_{k(j)} \ge \ell_j \text{ for } j \in F_u, \mathbf{x} \ge \mathbf{0}, \mathbf{y} \ge \mathbf{0}\}, \tag{$\mathrm{LP}_\ell(F_d, F_u)$}$$

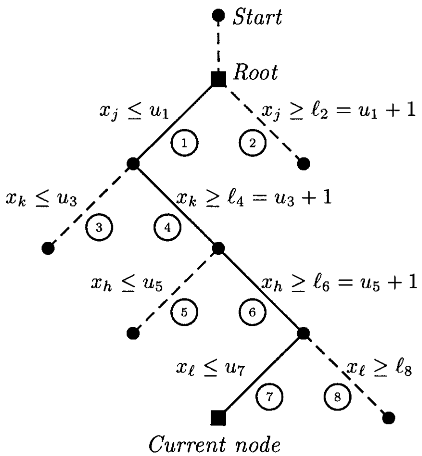

Fig. 6. The Branch-and-Bound path $P = F_d \cup F_u$.

where $(\mathbf{A}, \mathbf{D}, \mathbf{b})$ is the matrix consisting of the original constraints of (MIP) plus all global cuts that have been generated (and retained) so far. The slack or surplus variables of those cuts must (normally) be included into the set of flow variables. In actual computation a bounded variable version of the simplex algorithm is used to solve $(\mathrm{LP}_\ell(F_d, F_u))$. For the purpose of our analysis, we may assume instead that all inequalities (11) are converted to equations, i.e., we add to $\mathbf{Ax} + \mathbf{Dy} = \mathbf{b}$ the equations

$$x_{k(j)} + s_j = u_j \quad \text{for } j \in F_d, \quad x_{k(j)} - t_j = \ell_j \quad \text{for } j \in F_u, \tag{12}$$

where $s_j \geq 0$ and $t_j \geq 0$. We call the resulting linear program $(\mathrm{LP}_\ell^=(F_d, F_u))$.

In terms of the inequalities $(\mathrm{FC}^=)$ and $(\mathrm{FC}^\#)$ it is easy to state a sufficient condition for a post-branching inequality to be globally valid. Assume for notational simplicity that $\mathbf{A}$ in $(\mathrm{LP}_\ell^=(F_d, F_u))$ has m rows and let $s = |F_d| + |F_u|$ be the number of equations (12).

Proposition 4. *For every* $\boldsymbol{\mu} = (\boldsymbol{\mu}^{\mathbf{A}}, \boldsymbol{\mu}^{DU}) \in \mathbb{R}^{m+s}$ *with* $\boldsymbol{\mu}^{\mathbf{A}} \in \mathbb{R}^m$ *and* $\boldsymbol{\mu}^{DU} = \mathbf{0} \in \mathbb{R}^s$ *the post-branching inequalities* $(FC^=)$ *and* $(FC^\#)$ *are globally valid.*

Proof. If $\boldsymbol{\mu}^{DU} = \mathbf{0}$ then from the derivation of $(\mathrm{FC}^=)$ and $(\mathrm{FC}^\#)$ it follows that both are valid inequalities for all mixed-integer solutions to $\mathbf{Ax} + \mathbf{Dy} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$. By assumption $\mathbf{Ax} + \mathbf{Dy} = \mathbf{b}$ consists only of the original equations of (MIP) plus some global cuts. ■

To translate the condition of the proposition to the basis-dependent cuts $(\mathrm{GM}^=)$ and $(\mathrm{GM}^\#)$ let $\mathbf{B_a}$ be any basis and $\mathbf{h}$ be the right-hand side vector of $(\mathrm{LP}_\ell^=(F_d, F_u))$. Denote by $\bar{\mathbf{h}} = \mathbf{B_a}^{-1}\mathbf{h}$ the transformed right-hand side and let $r \in \{1, \ldots, m\}$ be such that the associated basic variable $x_{k(r)}$ is an integer variable, but $\bar{h}_r \notin \mathbb{Z}$. The corresponding row in the reduced equation format then reads

$$x_{k(r)} + \sum_{j \in A} \bar{a}_j^r x_j + \sum_{j \in B} \bar{d}_j^r y_j + \sum_{j \in C} \bar{g}_j^r s_j + \sum_{j \in D} \bar{g}_j^r t_j = \bar{h}_r, \tag{13}$$

where A is the index set of the nonbasic $\mathbf{x}$ variables, B the set of nonbasic $\mathbf{y}$ variables, C and D the ones for the nonbasic variables s_j and t_j from (12), respectively.

Proposition 5. *If* $\mathbf{B_a}$ *is a basis for* $(LP_\ell^=(F_d, F_u))$ *such that in (13)* $C \cup D = \emptyset$ *or* $\bar{g}_j^r = 0$ *for all* $j \in C \cup D$*, then the post-branching cuts* $(GM^=)$ *and* $(GM^\#)$ *derived from (13) are global cuts.*

Proof. Let $\boldsymbol{\mu} = \mathbf{u}_r \mathbf{B_a}^{-1}$ where $\mathbf{u}_r \in \mathbb{R}^{m+s}$ is the *r-th* unit vector. It suffices to show that the condition of Proposition 4 is satisfied. Let $N_P \subseteq N$ be the indices of the integer variables along the path P from the root to the current node and I be index set of variables in the basis $\mathbf{B_a}$. Then $I = I_\mathbf{x} \cup I_\mathbf{y} \cup I_0 \cup I_1 \cup I_2$, where

$$I_\mathbf{x} = \{j \in N - N_P : x_j \text{ is basic}\},\ I_\mathbf{y} = \{j \in \{1, \ldots, p\} : y_j \text{ is basic}\}$$

$$I_0 = \{j \in N_P : x_j \text{ is basic, } s_k \text{ and } t_k \text{ are basic for all } k \in B_j\}$$

$$I_1 = \{j \in N_P : x_j \text{ is basic, } s_k \text{ or } t_k \text{ is nonbasic for some } k \in B_j\}$$

$$I_2 = \{j \in N_P : x_j \text{ is nonbasic, } s_k \text{ and } t_k \text{ are basic for all } k \in B_j\} \qquad (14)$$

and we have assumed that the total number of flow variables (including the slack or surplus variables for the global cuts already added) equals p. Consequently,

$$\mathbf{B_a} = \begin{pmatrix} \mathbf{B_x} & \mathbf{B_y} & \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{F}_1 & \pm\mathbf{I}_2 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I}_3 & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{F}_0 & \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I}_4 \end{pmatrix},$$

where $\mathbf{B_x}$, $\mathbf{B}_0$ and $\mathbf{B}_1$ are submatrices of $\mathbf{A}$ with columns in $I_\mathbf{x}$, I_0 and I_1, respectively and $\mathbf{B_y}$ is the submatrix of $\mathbf{D}$ with columns in $I_\mathbf{y}$. $\mathbf{I}_1$ is an identity matrix corresponding to I_1. The $\pm\mathbf{I}_i$ for $i \in \{2, \ldots, 4\}$ are signed identity matrices corresponding to the slack/surplus variables in (12). E.g. in $\pm\mathbf{I}_2$ a plus sign is used if the corresponding basic variable is a s_j variable and the minus sign is used if it is a t_j variable. The $\mathbf{F}_i$ for $i \in \{0, 1, 3\}$ are matrices having exactly one entry equal to 1 per row. $\mathbf{O}$ are compatibly dimensioned matrices of zeros only. Let

$$\mathbf{B_{xy}} = (\mathbf{B_x} \quad \mathbf{B_y} \quad \mathbf{B}_0), \quad \mathbf{G}_0 = (\mathbf{O} \quad \mathbf{O} \quad \mathbf{F}_0).$$

It follows that $\mathbf{B_{xy}}$ is $m \times m$ and $\det \mathbf{B_a} = \pm \det \mathbf{B_{xy}}$. Consequently, $\mathbf{B_a}$ and its inverse are

$$\mathbf{B_a} = \begin{pmatrix} \mathbf{B_{xy}} & \mathbf{B}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{F}_1 & \pm\mathbf{I}_2 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I}_3 & \mathbf{O} \\ \mathbf{G}_0 & \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I}_4 \end{pmatrix},$$

$$\mathbf{B_a}^{-1} = \begin{pmatrix} \mathbf{B_{xy}}^{-1} & -\mathbf{B_{xy}}^{-1}\mathbf{B}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{K}_2 & \pm\mathbf{I}_2 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I}_3 & \mathbf{O} \\ \mathbf{K}_1 & \mathbf{K}_3 & \mathbf{O} & \mathbf{O} & \pm\mathbf{I}_4 \end{pmatrix},$$

where $\mathbf{K}_1 = \mp\mathbf{I}_4\mathbf{G}_0\mathbf{B_{xy}}^{-1}$, $\mathbf{K}_2 = \mp\mathbf{I}_2\mathbf{F}_1$ and $\mathbf{K}_3 = \pm\mathbf{I}_4\mathbf{G}_0\mathbf{B_{xy}}^{-1}\mathbf{B}_1$. Let $\mathbf{R_x}$ and $\mathbf{R}_1$ be the submatrices of $\mathbf{A}$ of the nonbasic x_j with $j \in N - N_P$ and $j \in I_2$, respectively, $\mathbf{R_y}$ be the submatrix of $\mathbf{D}$ of the nonbasic flow variables and denote by $\mathbf{R_a}$ the matrix of all nonbasic columns of $(\mathrm{LP}_\ell^{=}(F_d, F_u))$. We calculate

$$\mathbf{R_a} = \begin{pmatrix} \mathbf{R_x} & \mathbf{R_1} & \mathbf{R_y} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I_1} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathrm{F_3} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{pmatrix},$$

$$\mathbf{B_a^{-1}R_a} = \begin{pmatrix} \mathbf{B_{xy}^{-1}R_x} & \mathbf{B_{xy}^{-1}R_1} & \mathbf{B_{xy}^{-1}R_y} & \mathbf{B_{xy}^{-1}B_1}(\mp\mathbf{I_1}) \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \pm\mathbf{I_1} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K_2}(\pm\mathbf{I_1}) \\ \mathbf{O} & \pm\mathbf{I_3F_3} & \mathbf{O} & \mathbf{O} \\ \mathbf{K_1R_x} & \mathbf{K_1R_1} & \mathbf{K_1R_y} & \mathbf{K_3}(\pm\mathbf{I_1}) \end{pmatrix}.$$

Let $\mathbf{v}_r \in \mathbb{R}^m$ be the *r-th* unit vector. Thus

$$\boldsymbol{\mu} = \mathbf{u}_r\mathbf{B_a^{-1}} = (\mathbf{v}_r\mathbf{B_{xy}^{-1}} \quad -\mathbf{v}_r\mathbf{B_{xy}^{-1}B_1} \quad \mathbf{O} \quad \mathbf{O}) = (\boldsymbol{\mu}^{\mathbf{A}} \quad \boldsymbol{\mu}^{DU}).$$

Thus if $C \cup D = \varnothing$ then $\boldsymbol{\mu}^{DU} = \mathbf{0}$. Otherwise, from the calculation of $\mathbf{B_a^{-1}R_a}$ and the assumption that $\bar{g}_j^r = 0$ for all $j \in C \cup D$ it follows that $\mathbf{v}_r\mathbf{B_{xy}^{-1}B_1}(\mp\mathbf{I_1}) = \mathbf{0}$. Hence again $\boldsymbol{\mu}^{DU} = \mathbf{0}$ and the assertion follows from Proposition 4. ■

3.3 The mixed zero-one case

Let us now consider the important special case of (MIP) where the integer variables are all zero-one variables. For the analysis of this case we may assume that we have q "original" zero-one variables x_j and that the formulation of (MIP) contains the equations

$$x_j + x_{q+j} = 1 \quad \text{for } j = 1, \ldots, q. \tag{15}$$

Since the "complement" variables x_{q+j} must be zero-one valued as well we thus have $n = 2q$ integer variables. Like in the general case we denote by P the path from the root of the search tree to the current node. Thus (11) becomes

$$x_{k(j)} \leq 0 \quad \text{for all } j \in F_d, \quad x_{k(j)} \geq 1 \quad \text{for all } j \in F_u,$$

where $F_d \subseteq \mathscr{L}$ and $F_u \subseteq \mathscr{L}$ are the down and the up branches of the path $P = F_d \cup F_u$. In a mixed zero-one program we can rule out "contradictory" as well as "repeated" branching on the same variable. It follows that, by choosing an appropriate labeling, we can assume without loss of generality that the set N_P of integer variables along the path P satisfies $N_P = P$ and $B_j = \{j\}$ for all $j \in P$. Like in the general case, we assume for our analysis that all branching equations (12) have been added explicitly to form $(\mathrm{LP}_\ell^=(F_d, F_u))$ and that branching is done on the original variables x_j rather than on their complements x_{q+j}.

Corollary 1 (Balas et al. [5]). *If the integer variables of (MIP) are all zero-one variables and* $\mathbf{B_a}$ *is any basis for* $(LP_\ell^=(F_d, F_u))$ *such that either* x_j *or* x_{q+j}*, see (15), for* $j \in P$ *is in the basis (but not both), then all post-branching cuts* $(GM^=)$ *and* $(GM^\#)$ *are global cuts.*

Proof. If the set I_1, see (14), is empty, then Proposition 5 applies and the assertion follows. So suppose that $I_1 \neq \emptyset$. Since by assumption either x_j or x_{q+j} with $j \in P$ is in the basis, $\mathbf{B_a}$ contains a unit row corresponding to the equation (15). Since $I_1 \neq \emptyset$, $\mathbf{B_a}$ contains another unit row identical to the first one (corresponding to the up or down branch equation (12)), which contradicts the nonsingularity of $\mathbf{B_a}$. ■

If e.g. a bounded variable simplex algorithm is used to optimize the linear program $(\mathrm{LP}_\ell^=(F_d, F_u))$, then the "complementarity" condition of the corollary is automatically satisfied, because neither (12) nor (15) is needed to form the basis. In the mixed zero-one case, the "current" basis, e.g. an optimal basis of $(\mathrm{LP}_\ell^=(F_d, F_u))$, is simply a feasible, but nonoptimal basis to the original linear program augmented by some global cuts (where $x_{k(j)}$ for $j \in F_d$ and $x_{q+k(j)}$ for $j \in F_u$ are nonbasic at value zero). Any basis of that constraint set works for the derivation of global cuts $(\mathrm{GM}^=)$ and$(\mathrm{GM}^\#)$ and thus their designation as "lifted cuts" [5] is a misnomer. However, as the following example shows, the condition of the corollary is necessary when (12) and (15) are added explicitly to the linear program even in the zero-one case.

Example 8. Consider the mixed zero-one linear program (MIP_{01}) from [5]

$$\begin{array}{lrcrcrcrcl} \min & 3x_1 & + & x_2 & + & 3x_3 & + & 4y_1 & & \\ \textit{subject to} & 2x_1 & + & 3x_2 & + & x_3 & + & y_1 & = & 4 \end{array}$$

where $x_i \in \{0, 1\}$ for $i = 1, 2, 3$, i.e., $q = 3$, and $y_1 \geq 0$. An optimal solution to the linear programming relaxation of (MIP_{01}) is given by $x_1 = \frac{1}{2}$, $x_2 = 1$ and $x_3 = y_1 = 0$. Suppose that we branch on x_1 and consider the problem on the up branch $x_1 \geq 1$. Introducing the equations (12) and (15) like we did in the above analysis

$$x_1 + x_4 = 1, \quad x_2 + x_5 = 1, \quad x_3 + x_6 = 1, \quad x_1 - t_1 = 1$$

one calculates that the matrix corresponding to x_1, x_2, x_4, x_5 and x_6 is an optimal basis $\mathbf{B_a}$ to the resulting relaxed linear program which violates the condition of the corollary, because both x_1 and x_4 are basic. The corresponding reduced equation format is given by

$$\begin{array}{ccccccccccc} x_0 & & & -\frac{8}{3}x_3 & & & & -\frac{11}{3}y_1 & -\frac{7}{3}t_1 & = & 3\frac{2}{3} \\ & x_1 & & & & & & & -t_1 & = & 1 \\ & & x_2 & +\frac{1}{3}x_3 & & & & +\frac{1}{3}y_1 & +\frac{2}{3}t_1 & = & \frac{2}{3} \\ & & & & x_4 & & & & +t_1 & = & 0 \\ & & & -\frac{1}{3}x_3 & & +x_5 & & -\frac{1}{3}y_1 & -\frac{2}{3}t_1 & = & \frac{1}{3} \\ & & & x_3 & & & +x_6 & & & = & 1 \end{array}$$

Consequently, the application of $(\mathrm{GM}^{\#})$ to x_2 gives the cut $\frac{1}{2}x_3 + \frac{1}{2}y_1 + t_1 \geq 1$, i.e.,

$$2x_1 + x_3 + y_1 \geq 4$$

which is globally invalid because it cuts off the feasible solution $x_2 = x_3 = 1$, $x_1 = y_1 = 0$ to (MIP_{01}). This is independent upon whether or not t_1 is considered an integer variable and demonstrates again the basis dependency of the cut $(\mathrm{GM}^{\#})$. Indeed, the example shows that the application of $(\mathrm{GM}^{\#})$ may produce globally invalid cuts depending upon the way the linear relaxation of (MIP_{01}) and its subproblems in branch-and-cut are *formulated* and solved *algorithmically*. Moreover, in this case $\boldsymbol{\mu} = \mathbf{u}_2\mathbf{B}_{\mathbf{a}}^{-1} = (\frac{1}{3}\ \ 0\ \ 0\ \ 0\ \ -\frac{2}{3})$ shows that the condition of Proposition 4 is violated. ■

Combining Proposition 5 and Corollary 1 gives an evident sufficient condition for post-branching cuts to be globally valid whenever (MIP) has zero-one as well as general integer variables.

3.4 Using post-branching cuts in B&C

The condition of Proposition 5 is easy to verify in actual computation and thus the generation of global cuts using e.g. $(\mathrm{GM}^{\#})$ becomes possible even after branching has occurred. Moreover, if the condition of the proposition is "nearly" satisfied, e.g. if $|C \cup D| = 1$ in (13), then it may be possible to move from the current basis to a "nearby" infeasible basis for $(\mathrm{LP}_{\ell}^{=}(F_d, F_u))$ which satisfies the condition. Denote e.g. by $\bar{\mathbf{g}}_j$ the transformed column of some slack s_j or surplus t_j variable with $\bar{g}_j^r \neq 0$ in (13). Let s be the row for which

$$\min\left\{\frac{\bar{h}_i}{|\bar{g}_j^i|} : \bar{g}_j^i < 0, 1 \leq i \neq r \leq m\right\}$$

is attained. Then pivoting the s_j or t_j variable into the basis from row s produces typically an infeasible basis and a basic solution "near" to the original one from which a global cut may be derived. Note that the derivation of e.g. $(\mathrm{GM}^{\#})$ requires neither that the basis be feasible nor that it be optimal. Evidently, this opens up many strategic plays for experimentation. Rather than discussing such heuristic choices in detail we illustrate the principle by three examples.

Example 9. Consider the problem from Example 7 along the down branch $x_2 \leq 3$, i.e.,

$$\begin{array}{lllllllll}
x_1 & & & + \frac{1}{8}x_4 & - \frac{3}{8}x_5 & - \frac{1}{8}x_6^d & = & \frac{1}{2} \\
& x_2 & & & & + x_6^d & = & 3 \\
& & x_3 & - \frac{3}{8}x_4 & + \frac{1}{8}x_5 & - \frac{5}{8}x_6^d & = & \frac{1}{2}
\end{array}$$

which produced globally invalid cuts. In row 1 the condition of Proposition 5 is violated. Pivoting the variable x_6^d into the basis from row 3 we obtain the equation format

$$\begin{array}{rrrrrrrrrrr} x_1 & & - & \frac{1}{5}x_3 & + & \frac{1}{5}x_4 & - & \frac{2}{5}x_5 & & & = & \frac{2}{5} \\ & x_2 & + & \frac{8}{5}x_3 & - & \frac{3}{5}x_4 & + & \frac{1}{5}x_5 & & & = & 3\frac{4}{5} \\ & & - & \frac{8}{5}x_3 & + & \frac{3}{5}x_4 & - & \frac{1}{5}x_5 & + & x_6^d & = & -\frac{4}{5} \end{array}$$

which – in this simple demonstration of the principle – is the first equation format amended by the violated upper bound equation. Now row 1 satisfies the condition of Proposition 5 and we can use it to generate a global cut. Note also that row 3 can be used to generate a global cut since the nonnegativity of the basic variables is not used in the derivation of ($GM^{\#}$). On the other hand, consider (MIP_{01}) of Example 8. Pivoting the variables x_4 out of and t_1 into the basis, the optimality of the basis is lost. However, now the condition of the corollary is satisfied and applying ($GM^{\#}$) to the changed row corresponding to x_2 one gets the global cut

$$-x_1 + x_3 + y_1 \geq 1$$

for (MIP_{01}). Another simple trick that may sometimes work to get globally valid inequalities goes as follows. Using the equations (12) for $j \in C \cup D$ we rewrite (13) as

$$x_{k(r)} + \sum_{j\in A} \bar{a}_j^r x_j + \sum_{j\in B} \bar{d}_j^r y_j + \sum_{j\in C} (-\bar{g}_j^r) x_{k(j)} + \sum_{j\in D} \bar{g}_j^r x_{k(j)}$$

$$= \bar{h}_r - \sum_{j\in C} \bar{g}_j^r u_j + \sum_{j\in D} \bar{g}_j^r \ell_j \tag{16}$$

and apply ($GM^{\#}$) to (16). This operation corresponds to pivoting all $x_{k(j)}$ with $j \in C \cup D$ out of the basis and the corresponding s_j and t_j into the basis. Row r in the new basis satisfies the condition of Proposition 5, see (16), and thus global validity of ($GM^{\#}$) follows. However, the inequality will typically cease to be a cut. If e.g. along the down branch of the problem of Example 7 we eliminate $x_6^d = 3 - x_2$ from the equation format, then from the x_1 row we find $2x_1 + x_2 + x_3 \geq 5$ which is a global cut, while from the x_3 row we find $3x_1 + 4x_2 + 4x_3 \geq 14$ which is globally valid, but not cutting. In large-scale mixed-integer programs it is very likely that the application of heuristic arguments of this kind will in general help to find additional global cuts for the problem even in the post-branching phase. Moreover, the necessary calculations can evidently be streamlined to make them efficient. ■

Post-branching cuts that violate the condition of Proposition 4 are in general globally invalid and have to be "lifted" or modified in some fashion to make them globally valid for (MIP). There are several ways of how that can be done, see e.g. [52], [53], [58], [60], [61], [62], [63] for related ideas, but at the expense of additional computations and without a guarantee that a "cut" be obtained in all cases. Here is one way of doing so. Suppose for simplicity that $C = \{j\}$ and $D = \emptyset$ in (13) and denote by π_j (ρ_j) for $j \in A$ (for $j \in B$) the coefficients of the inequality obtained from ($GM^{\#}$) by assuming that $s_j = 0$. The corresponding inequality then is valid for (MIP) with the additional constraints that $x_{k(j)} = u_j$ for all $j \in F_d$ and $x_{k(j)} = \ell_j$ for all $j \in F_u$. Define

$$\chi(x_{k(j)}) = \min\Bigg\{ \sum_{j\in A} \pi_j x_j + \sum_{j\in B} \rho_j y_j : x_{k(r)} + \sum_{j\in A} \bar{a}_j^r x_j + \sum_{j\in B} \bar{d}_j^r y_j = \bar{h}_r - \bar{g}_j^r u_j + \bar{g}_j^r x_{k(j)},$$

$$x_{k(r)}, x_j \in \mathbb{Z}_+ \text{ for } j \in A, y_j \in \mathbb{R}_+ \text{ for } j \in B \Bigg\},$$

which is a one-rowed parametric mixed integer program. Let $\beta \geq 0$, γ, $\delta \in \mathbb{R}$ be such that

$$\beta\chi(x_{k(j)}) + \gamma x_{k(j)} \geq \delta$$

for all $x_{k(j)} \in \mathbb{Z}_+$. In this construction one chooses $\beta \geq 0$, γ and δ such that equality is obtained for at least two different points $(x_{k(j)}, \chi(x_{k(j)}))$ with $x_{k(j)} \in \mathbb{Z}_+$. Since $\beta \geq 0$ it follows that the inequality

$$\beta\left(\sum_{j\in A} \pi_j x_j + \sum_{j\in B} \rho_j y_j \right) + \gamma x_{k(j)} \geq \delta$$

is globally valid for (MIP) and a cut if $\gamma u_j < \delta$.

If $|C \cup D| > 1$, anyone (or all) of the inequalities thus obtained for the single variable s_j (and that is violated by the present solution) is modified likewise to include the remaining branching variables corresponding to $k \in C \cup D - j$ with $\bar{g}_k^r \neq 0$ one by one. If the problem is sparse, i.e., if the number of nonzero elements in (13) is small, then the necessary calculations can be carried out efficiently, but any measure of the mathematical "quality" of the *possible* cuts obtained this way cannot be guaranteed.

Evidently the choice of the multipliers $\boldsymbol{\mu}$, see (8), in the derivation of (GM$^\#$) is only one of many possibilities and leaves lot of room for experimentation. In Ceria et al. [12] cuts using multipliers of the form $\boldsymbol{\mu} = \mathbf{u}\mathbf{B}^{-1}$ with $\mathbf{u} \in \mathbb{Z}^m$ are discussed, which is in the spirit of earlier works on improving Gomory's fractional and all-integer cuts, see e.g. [29], [54], [82] and others.

4 Further remarks

The above discussion has shown that classical valid inequalities for mixed-integer programming can, but must be used with caution in branch-and-cut. However, one aspect that we have not stressed so far is the precision of calculation. Like most other authors we have used in our small examples *exact* arithmetic and thus avoided the difficulties that arise from the use of floating point operations. To illustrate the point we wish to make consider the problem

$$\begin{array}{lrcrcrcl} & \max & 4x_1 & + & 7x_2 & + & 6x_3 & \\ \textit{subject to} & & 3x_1 & + & 7x_2 & + & 7x_3 & = 14 \end{array}$$

where $x_i \geq 0$ are integer variables for $i = 1, 2, 3$. The optimum solution of the linear programming relaxation is $x_1 = 4\frac{2}{3}$, $x_2 = x_3 = 0$. Suppose that we carry out all calculations with a precision of three positions after the decimal point. We get the equation

$$x_1 + 2.333x_2 + 2.333x_3 = 4.667$$

from which we derive e.g. the cut ($\mathrm{GM}^{\#}$) in the same precision to be

$$0.499x_2 + 0.499x_3 \geq 1$$

which cuts off the feasible solution $x_1 = 0$, $x_2 = x_3 = 1$. This is due to the finite precision of the calculations. All (commercial) LP solvers to date use floating point calculations with finite precision. As a result errors occur in the calculation of a "solution": integrality in floating point calculations is checked *via* tolerances. The calculation of the coefficients that are subsequently used to derive the cuts is also prone to errors. With respect to cut generation the precision-related errors have two possible effects:

1. *one may lose the optimum by simply cutting it off with an invalid cut and*
2. *the search tree may become "short" or even "very short" for the same reason.*

Undeniably, invalid cuts may be produced as a result of the finite precision of the calculations. Given the derivation of e.g. ($\mathrm{GM}^{\#}$) we do not see how numerical analysis might help to set "tolerances" in a mathematical correct way to avoid the generation of invalid cuts. "Rounding up", like we could always do in ($\mathrm{GM}^{\#}$), e.g. above by replacing 0.499 by 0.5 because $x_2 \geq 0$ and $x_3 \geq 0$, may do the job sometimes, but at the expense of making a possibly weak (and dense) cut even weaker (and denser). It appears that the only satisfactory way to resolve this problem of precision is to build LP solvers that use *exact arithmetic* because the classical cuts use the properties of the field of rational numbers in their derivation in a decisive way.

The recent literature sweeps these precision related problems under the rug using cavalier statements like LP solvers "are more robust" [5] and, in the same vein, "Times have changed" [10]. In [15] no mention to this effect is to be found at all. By contrast, in the earlier applications exact arithmetic was used, see e.g. [19] and [76]. Moreover, part of the recently reported success with ($\mathrm{GM}^{\#}$)is attributed to the fact that cuts are generated in "rounds" rather than one at a time. The latter is computationally the most unattractive strategic choice (and used only to prove convergence theorems.) Indeed, many of the strategic choices reported in e.g. [5] were tried e.g. in [76] – with reportedly very bad results for problem sizes that we consider "small" by today's standards.

For classical cuts a handy formula is available that – given a basis for the associated linear program – permits one to find the cut easily, i.e., here the separation problem poses no specific problems, at least *prior* to branching. This is different from cuts based on facet-defining inequalities where problem specific separation algorithms must be invoked. While the latter cuts are best possible cuts, not much is known about the mathematical properties of classical cuts. This should, however, not discourage a problem solver from using them – provided they are used in a mathematical correct way – since in branch-and-cut one can always default to branching when "tailing off" occurs, see e.g. [68], [69]. The use of classical cuts does in no way preclude the use of facet-defining cuts in the solution process – if they are known for the particular problem to be optimized (or substructures of it). In other words, classical cuts and cuts obtained from facet-defining inequalities can perfectly co-exist; see [10] and [15] – *modulo* our reservations about precision.

References

[1] Applegate D, Bixby R, Chvátal V, Cook W (1998) On the solution of traveling salesman problems. Documenta Mathematica: Extra Vol. ICM, III 645–656
[2] Balas E (1963) Programare liniara cu variabile bivalente. Proceedings Third Scientific Session on Statistics, Bucharest. French translation (1964) Un algorithme additive pour la résolution des programmes linéaires en variables bivalentes. Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences 258:3817–3820. English translation (1965) An additive algorithm for solving linear programs with zero-one variables. Operations Research 13:517–546
[3] Balas E (1971) Intersection cuts – a new type of cutting planes for integer programming. Operations Research 19:19–39
[4] Balas E (1975) Facets of the knapsack polytope. Mathematical Programming 8:146–164
[5] Balas E, Ceria S, Cornuéjols G, Natraj N (1996) Gomory cuts revisited. Operations Research Letters 19:1–9
[6] Balinski M, Spielberg K (1969) Methods for integer programming: algebraic, combinatorial and enumerative. in Aronofsky (ed), Progress in Operations Research: Relationship between Operations Research and the Computer, Vol. III, Wiley, New York 195–292
[7] Barany I, Van Roy T, Wolsey L (1984) Strong formulations for multi-item capacitated lot-sizing. Management Science 30:1255–1261
[8] Beale M (1958) A method of solving linear programming problems when some but not all of the variables must take integral values. Statistical Research Group, Technical Report No. 9, Princeton U, Princeton, N.J.
[9] Beale M (1965) Survey of integer programming. Operational Research Quarterly 16:219–228
[10] Bixby R, Fenelon M, Gu Z, Rothberg E, Wunderling R (1999) MIP: theory and practice closing the gap. Research Report, ILOG Inc., Lake Tahoe, Incline Village, Nevada
[11] Ceria S, Cordier C, Marchand H, Wolsey L (1998) Cutting planes for integer programs with general integer variables. Mathematical Programming 81:201–214
[12] Ceria S, Cornuéjols G, Dawande M (1995) Combining and strengthenig Gomory cuts. in Balas and Clausen (eds), Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science No. 920, Springer-Verlag, Berlin 438–451
[13] Cho D, Johnson E, Padberg M, Rao MR (1983) On the uncapacitated plant location problem I: valid inequalities and facets. Mathematics of Operations Research 8:579–589
[14] Cho D, Padberg M, Rao MR (1983) On the uncapacitated plant problem II: lifting theorems and facets. Mathematics of Operations Research 8:590–612
[15] Cordier C, Marchand H, Laundy R, Wolsey L (1999) bc-opt: a branch-and-cut code for mixed integer programs. Mathematical Programming 86:335–353
[16] Crowder H, Padberg M (1980) Solving large-scale symmetric traveling salesman problems to optimality. Management Science 26:495–509
[17] Crowder H, Johnson E, Padberg M (1983) Solving large-scale zero-one linear programming problems. Operations Research 31:803–834
[18] Dantzig G, Fulkerson DR, Johnson S (1954) Solution of a large-scale traveling salesman problem. Operations Research 2:393–410
[19] Fleischmann B (1970) Duale und Primale Schnitthyperebenenverfahren in der ganzzahligen linearen Optimierung, Doctoral Dissertation, Mathematik, Universität Hamburg
[20] Fiorot J, Gondran M (1969) Résolution des systèmes linéaires en nombres entiers. Bulletin de la Direction des Etudes et Recherches E.D.F., Series C 2:65–116

[21] Forrest J (1979) Current state of computer codes for discrete optimization. in Hammer, Johnson and Korte (eds), Discrete Optimization II, Annals of Discrete Mathematics 5:271–274
[22] Garfinkel R, Nemhauser G (1972) Integer Programming. Wiley & Sons, New York
[23] Geoffrion A (1969) An improved implicit enumeration approach for integer programming. Operations Research 17:437–454
[24] Geoffrion A, Marsten R (1972) Integer programming algorithms: a framework and state-of-the-art survey. Management Science 18:465–491
[25] Glover F (1968) A new foundation for a simplified primal integer programming algorithm. Operations Research 16:727–740
[26] Gomory R (1958) An algorithm for integer solutions to linear programs. Princeton-IBM TR No. 1, Nov. 1958; abstracted in Outline of an algorithm for integer solutions to linear programs. Bull. Amer. Math. Soc. 64:275–278; published in Graves and Wolfe (eds), Recent Advances in Mathematical Programming, McGraw-Hill, New York 1963:269–302
[27] Gomory R (1960) All-integer integer programming algorithm. IBM Research Center, Yorktown Heights, RC-189, January 1960 ; published in Muth and Thompson (eds), Industrial Scheduling, Prentice-Hall, Englewoods Cliffs, NJ 1963:193–206
[28] Gomory R (1960) An algorithm for the mixed integer problem. TR RM-2597, The Rand Corporation, 1960
[29] Gondran M (1973) Un outil pour la programmation en nombres entiers: la méthode des congruences decroissantes. RAIRO V-3:35–54
[30] Grötschel M, Holland O (1987) A cutting-plane algorithm for minimum perfect 2-matching. Computing 39:327–344
[31] Grötschel M, Jünger M, Reinelt G (1984) A cutting plane algorithm for the linear ordering problem. Operations Research 32:1195–1220
[32] Grötschel M, Lovász L, Schrjiver A (1988) Geometric algorithms and combinatorial optimization. Springer Verlag, Berlin
[33] Grötschel M, Padberg M (1979) On the symmetric traveling salesman problem I: inequalities, II: lifting theorems and facets. Mathematical Programming 16:265–302
[34] Grötschel M, Padberg M (1999) Die optimierte Odyssee. Spektrum der Wissenschaft, April 1999; reprinted in special issue Wissenschafliches Rechnen (Scientific Computing), October 1999; translated into French L'Odyssée abregée. Pour la Science, August 1999
[35] Haldi J, Isaacson L (1965) A computer code for integer solutions to linear programs. Operations Research 13:946–959
[36] Hoffman K, Padberg M (1985) LP-based combinatorial problem solving. Annals of Operations Research 4:145–194
[37] Hoffman K, Padberg M (1991) Improving the LP-representation of zero-one linear programming problems for branch-and-cut. ORSA Journal on Computing 3:121–134
[38] Hoffman K, Padberg M (1993) Solving airline crew scheduling problems by branch-and-cut. Management Science 39:657–682
[39] Jeroslow R (1979) An introduction to the theory of cutting planes. Annals of Discrete Mathematics 5:71–95
[40] Johnson E (1974) The group problem for mixed integer programming. Mathematical Programming Study 2:137–179
[41] Jünger M, Reinelt G, Thienel S (1995) Practical problem solving with cutting plane algorithms in combinatorial optimization. in Cook, Lovász and Seymour (eds), Combinatorial Optimization, DIMACS Series in Discrete Mathematics and Computer Science, AMS
[42] Krolak P (1969) Computational results of an integer programming algorithm. Operations Research 17:743–749
[43] Knuth D (1961) Minimizing drum latency time. JACM 8:119–150
[44] Land A, Doig A (1960) An automatic method for solving discrete programming problems. Econometrica 28:497–520
[45] Land A, Powell S (1979) Computer codes for problems of integer programming. in Hammer, Johnson and Korte (eds), Discrete Optimization II, Annals of Discrete Mathematics 5:221–269
[46] Leung J, Magnanti T (1989) Valid inequalities and facets of the capacitated plant location problem. Mathematical Programming 44:271–291
[47] Leung J, Magnanti T, Vachani R (1989) Facets and algorithms for capacitated lot sizing. Mathematical Programming 45:331–359

[48] Magnanti T, Vachani R (1990) A strong cutting plane algorithm for production scheduling with changeover costs. Operations Research 38:456–473
[49] Marchand H, Wolsey L (1998) Aggregation and mixed-integer programming rounding to solve MIPs. CORE Discussion Paper No. 9839, Université Catholique, Louvain-la-Neuve
[50] Marchand H, Martin A, Weismantel R, Wolsey L (1999) Cutting planes in integer and mixed-integer programming. CORE Discussion Paper No. 9953, Université Catholique, Louvain-la-Neuve
[51] Markowitz H, Manne A (1957) On the solution of discrete programming problems. Econometrica 25:84–110
[52] Martin A (1998) Integer Programs with Block Structure. Habilitations-Schrift, Technical University, Berlin
[53] Martin A, Weismantel R (1998) The intersection of knapsack polyhedra and extensions. in Bixby, Boyd and Rios-Mercado (eds), Integer Programming and Combinatorial Optimization, Proceedings of the *6th* IPCO Conference 243–256
[54] Martin G (1963) An accelerated Euclidean algorithm for integer linear programming. in Graves and Wolfe (eds), Recent Advances in Mathematical Programming, McGraw-Hill, New York 311–318
[55] Meyer R (1974) On the existence of optimal solutions to integer and mixed integer programming problems. Mathematical Programming 7:223–235
[56] Miliotis P (1978) Using cutting planes to solve symmetric traveling salesman problems. Mathematical Programming 15:177–188
[57] Nemhauser G, Wolsey L (1988) Integer and Combinatorial Optimization, Wiley & Sons, New York
[58] Padberg M (1973) On the facial structure of set packing polyhedra. Mathematical Programming 5:199–251
[59] Padberg M (1974) Unpublished lecture notes, 1974; see also A note on the convergence of integer programming algorithms using intersection cuts. Working paper, GSIA, Carnegie-Mellon U., 1970
[60] Padberg M (1975) A note on zero-one programming. Operations Research 23:833–837
[61] Padberg M (1977) On the complexity of set packing polyhedra. Annals of Discrete Mathematics 1:421–434
[62] Padberg M (1979) Covering, packing and knapsack problems. Annals of Discrete Mathematics 4:265–281
[63] Padberg M (1980) $(1,k)$-configurations and facets for packing problems. Mathematical Programming 18:94–99
[64] Padberg M (1989) The Boolean polytope: some characteristics, facets and relatives. Mathematical Programming 45:139–172
[65] Padberg M (1999) Linear Optimization and Extensions, 2^{nd} ed., Springer Verlag, Heidelberg
[66] Padberg M, Rao MR (1982) Odd minimum cut sets and b-matchings. Mathematics of Operations Research 7:67–80
[67] Padberg M, Rijal M (1996) Location, Scheduling, Design and Integer Programming, Kluwer Academic Publishers, Boston
[68] Padberg M, Rinaldi G (1987) Optimization of a 532-city symmetric traveling salesman problem by branch-and-cut. Operations Research Letters 6:1–7
[69] Padberg M, Rinaldi G (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review 33:60–100
[70] Padberg M, Wolsey L (1983) Trees and cuts. Annals of Discrete Mathematics 17:511–517
[71] Padberg M, Van Roy T, Wolsey L (1985) Valid linear inequalities for fixed charge problems. Operations Research 33:842–861
[72] Pochet Y (1988) Valid inequalities and separation for economic lot sizing. Operations Research Letters 7:109–116
[73] Pochet Y, Wolsey L (1988) Lot-size models with backlogging: Strong formulations and cutting planes. Mathematical Programming 40:317–335
[74] Salkin H (1975) Integer Programming, Addison-Wesley, Reading, MA
[75] Spielberg K (1979) Enumerative methods in integer programming. in Hammer, Johnson and Korte (eds), Discrete Optimization II, Annals of Discrete Mathematics 5:139–183
[76] Story A, Wagner H (1963) Computational experience with integer programming for job-shop scheduling. in Muth and Thompson (eds), Industrial Scheduling, Prentice-Hall, Englewoods Cliffs, NJ 207–219

[77] Taha H (1975) Integer Programming: Theory, Applications and Computations, Academic Press, New York
[78] Trauth Jr C, Woolsey R (1969) Integer linear programming: a study in computational efficiency. Management Science 15:481–493
[79] Van Roy T, Wolsey L (1987) Solving mixed 0-1 problems by automatic reformulation. Operations Research 35:45–57
[80] Wagner H, Giglio R, Glaser R (1964) Preventive maintenance scheduling by mathematical programming. Management Science 10:316–334
[81] White W (1961) On Gomory's mixed integer algorithm. Senior Thesis, Princeton University
[82] Wilson R (1967) Stronger cuts in Gomory's all-integer integer programming algorithm. Operations Research 15:155–157
[83] Wolsey L (1976) Facets and strong valid inequalities for integer programs. Operations Research 24:367–372
[84] Wolsey L (1989) Strong formulations for mixed integer programming: a survey. Mathematical Programming 45:173–191
[85] Wolsey L (1998) Integer Programming, Wiley, New York
[86] Young R (1968) A simplified primal (all-integer) integer programming algorithm. Operations Research 16:750–782
[87] Young R (1971) Hypercylindrically-deduced cuts in zero-one integer programs. Operations Research 19:1393–1405
[88] Zionts S (1974) Linear and Integer Programming, Prentice-Hall, Englewood Cliffs