Computational MIP: Selected Topics

Andrea Lodi
University of Bologna, Italy
andrea.lodi@unibo.it

January 13, 2010 @ Lunteren, The Netherlands

Outline, Assumptions and Notation

- We consider a general Mixed Integer Program in the form:

$$\min\{c^T x : Ax \geq b, x \geq 0,\ x_j \text{ integer},\ j \in \mathcal{I}\}$$

where matrix A does not have a special structure.

Outline, Assumptions and Notation

- We consider a general Mixed Integer Program in the form:

$$\min\{c^T x : Ax \geq b, x \geq 0,\ x_j \text{ integer},\ j \in \mathcal{I}\}$$

  where matrix A does not have a special structure.
- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

Outline, Assumptions and Notation

- We consider a general Mixed Integer Program in the form:

$$\min\{c^T x : Ax \geq b, x \geq 0,\ x_j \text{ integer},\ j \in \mathcal{I}\}$$

  where matrix A does not have a special structure.

- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

- Yesterday, we have mostly insisted on two different components of MIP solvers:
  1. branching and
  2. cutting

  and we also suggested that there is a strong relationship, very little understood, among them.

Outline, Assumptions and Notation

- We consider a general Mixed Integer Program in the form:

$$\min\{c^T x : Ax \geq b, x \geq 0,\ x_j \text{ integer},\ j \in \mathcal{I}\}$$

  where matrix A does not have a special structure.
- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

- Yesterday, we have mostly insisted on two different components of MIP solvers:
  1. branching and
  2. cutting

  and we also suggested that there is a strong relationship, very little understood, among them.

- Today, the talk discusses some new and sophisticated ideas for both components.

1: Traditional Branching

- The traditional way of partitioning the problem into sub-problems is the so-called variable branching. Pick a variable $x_j,\ j \in \mathcal{I}$ whose value x_j^* is fractional in the current LP relaxation and generate two sub-MIPs:

$$x_j \leq \lfloor x_j^* \rfloor \quad \bigvee \quad x_j \geq \lfloor x_j^* \rfloor + 1.$$

1: Traditional Branching

- The traditional way of partitioning the problem into sub-problems is the so-called variable branching. Pick a variable $x_j,\ j \in \mathcal{I}$ whose value x_j^* is fractional in the current LP relaxation and generate two sub-MIPs:

$$x_j \le \lfloor x_j^* \rfloor \quad \bigvee \quad x_j \ge \lfloor x_j^* \rfloor + 1. \tag{1}$$

- It is easy to see that "bad" decisions at early stages of the search, i.e., high levels in the tree, result in the exponential increase of the tree itself.

1: Traditional Branching

- The traditional way of partitioning the problem into sub-problems is the so-called variable branching. Pick a variable $x_j, \ j \in \mathcal{I}$ whose value x_j^* is fractional in the current LP relaxation and generate two sub-MIPs:

$$x_j \le \lfloor x_j^* \rfloor \quad \bigvee \quad x_j \ge \lfloor x_j^* \rfloor + 1. \tag{1}$$

- It is easy to see that "bad" decisions at early stages of the search, i.e., high levels in the tree, result in the exponential increase of the tree itself.

- Moreover, in the context of knapsack equality constraints with large coefficients and bounds branching on variables is not effective.
- This is the same for other types of MIPs, like symmetric ones.

1: Traditional Branching

- The traditional way of partitioning the problem into sub-problems is the so-called variable branching. Pick a variable $x_j,\ j \in \mathcal{I}$ whose value x_j^* is fractional in the current LP relaxation and generate two sub-MIPs:

$$x_j \leq \lfloor x_j^* \rfloor \quad \bigvee \quad x_j \geq \lfloor x_j^* \rfloor + 1. \tag{1}$$

- It is easy to see that "bad" decisions at early stages of the search, i.e., high levels in the tree, result in the exponential increase of the tree itself.

- Moreover, in the context of knapsack equality constraints with large coefficients and bounds branching on variables is not effective.
- This is the same for other types of MIPs, like symmetric ones.

- Finally, even for 0-1 combinatorial optimization problems it is often the case that fixing a variable to 1 is typically very strong <u>while</u> fixing it to 0 can have little or no effect for difficult instances.

1: Less Traditional Branching

- One recent line of research concerns branching on more complicated general disjunctions (in contrast with *elementary* ones, i.e., variable branching):

$$\alpha^T x \leq \alpha_0 \quad \bigvee \quad \alpha^T x \geq \alpha_0 + 1, \tag{2}$$

where α is an integer vector with 0-entries for variables $x_j,\ j \notin \mathcal{I}$ and α_0 is an integer scalar.

1: Less Traditional Branching

- One recent line of research concerns branching on more complicated general disjunctions (in contrast with *elementary* ones, i.e., variable branching):

$$\alpha^T x \leq \alpha_0 \quad \bigvee \quad \alpha^T x \geq \alpha_0 + 1, \qquad (2)$$

  where α is an integer vector with 0-entries for variables $x_j,\ j \notin \mathcal{I}$ and α_0 is an integer scalar.

- The key question is anyway: How should we select a disjunction?

1: Less Traditional Branching

- One recent line of research concerns branching on more complicated general disjunctions (in contrast with *elementary* ones, i.e., variable branching):

$$\alpha^T x \leq \alpha_0 \quad \bigvee \quad \alpha^T x \geq \alpha_0 + 1, \tag{2}$$

  where α is an integer vector with 0-entries for variables $x_j,\ j \notin \mathcal{I}$ and α_0 is an integer scalar.

- The key question is anyway: How should we select a disjunction?

- Typically, the set of disjunctions to be considered is limited a priori: either a candidate set of variables or a certain family of general disjunctions.
- From this limited set, one must choose the "best" disjunction by a given measure.

1: Less Traditional Branching

- One recent line of research concerns branching on more complicated general disjunctions (in contrast with *elementary* ones, i.e., variable branching):

$$\alpha^T x \le \alpha_0 \quad \bigvee \quad \alpha^T x \ge \alpha_0 + 1, \tag{2}$$

  where α is an integer vector with 0-entries for variables $x_j,\ j \notin \mathcal{I}$ and α_0 is an integer scalar.

- The key question is anyway: How should we select a disjunction?

- Typically, the set of disjunctions to be considered is limited a priori: either a candidate set of variables or a certain family of general disjunctions.
- From this limited set, one must choose the "best" disjunction by a given measure.

- The overall goal of any branching scheme is to reduce running time.
  As a proxy, most branching schemes try to maximize the (estimated) bound increase resulting from imposing the disjunction.

1: Selecting a Disjunction and Bilevel Programming

- The problem of selecting the disjunction whose imposition results in the largest bound improvement has a natural bilevel structure.

1: Selecting a Disjunction and Bilevel Programming

- The problem of selecting the disjunction whose imposition results in the largest bound improvement has a natural bilevel structure.
- This comes from the fact that the bound is computed by solving another optimization problem: roughly speaking, one is solving an optimization problem over the set of optimal solutions of another optimization problem.

1: Selecting a Disjunction and Bilevel Programming

- The problem of selecting the disjunction whose imposition results in the largest bound improvement has a natural bilevel structure.
- This comes from the fact that the bound is computed by solving another optimization problem: roughly speaking, one is solving an optimization problem over the set of optimal solutions of another optimization problem.

- The disjunction selection problem can sometimes be formulated as a bilevel program:
  - The upper-level variables can be used to model the choice of the disjunction (we'll see an example shortly).
  - The lower-level problem models the bound computation after the disjunction has been imposed.

1: Selecting a Disjunction and Bilevel Programming

- The problem of selecting the disjunction whose imposition results in the largest bound improvement has a natural bilevel structure.
- This comes from the fact that the bound is computed by solving another optimization problem: roughly speaking, one is solving an optimization problem over the set of optimal solutions of another optimization problem.

- The disjunction selection problem can sometimes be formulated as a bilevel program:
  - The upper-level variables can be used to model the choice of the disjunction (we'll see an example shortly).
  - The lower-level problem models the bound computation after the disjunction has been imposed.

- In strong branching, we are essentially solving the bilevel program by enumeration.

1: Selecting a Disjunction and Bilevel Programming

- The problem of selecting the disjunction whose imposition results in the largest bound improvement has a natural bilevel structure.
- This comes from the fact that the bound is computed by solving another optimization problem: roughly speaking, one is solving an optimization problem over the set of optimal solutions of another optimization problem.

- The disjunction selection problem can sometimes be formulated as a bilevel program:
  - The upper-level variables can be used to model the choice of the disjunction (we'll see an example shortly).
  - The lower-level problem models the bound computation after the disjunction has been imposed.

- In strong branching, we are essentially solving the bilevel program by enumeration.

- For general disjunctions, different authors have suggested different quality measures either circumventing the bilevel nature (Karamanov & Cornuéjols, and Cornuéjols, Liberti & Nannicini) or formulating it as a single level program (Mahajan & Ralphs).

1: Interdiction Branching

- At the price of solving the mentioned bilevel program one can be very sophisticated in selecting the disjunction. [Lodi, Ralphs, Rossi, Smriglio, 2009]

1: Interdiction Branching

- At the price of solving the mentioned bilevel program one can be very sophisticated in selecting the disjunction. [Lodi, Ralphs, Rossi, Smriglio, 2009]

- Consider, for the sake of simplicity, a binary integer program

$$\min\{c^T x \mid x \in \mathcal{P} \cap \mathbb{B}^n\},$$

where $c \in \mathbb{Q}^n$ is the objective function and $\mathcal{P}$ is a polyhedron.

1: Interdiction Branching

- At the price of solving the mentioned bilevel program one can be very sophisticated in selecting the disjunction. [Lodi, Ralphs, Rossi, Smriglio, 2009]

- Consider, for the sake of simplicity, a binary integer program

$$\min\{c^T x \mid x \in \mathcal{P} \cap \mathbb{B}^n\},$$

where $c \in \mathbb{Q}^n$ is the objective function and $\mathcal{P}$ is a polyhedron.

- We are interested in a set $S = \{i_1, \ldots, i_{|S|}\} \subseteq I^n = \{1, \ldots, n\}$, to impose the (clearly valid) multi-variable disjunction

$$x_{i_1} = 1 \vee (x_{i_2} = 1 \wedge x_{i_1} = 0) \vee \ldots \vee (x_{i_{|S|}} = 1 \wedge x_{i_1} = 0 \wedge \ldots \wedge x_{i_{|S|-1}} = 0) \vee \sum_{i \in S} x_i = 0.$$

1: Interdiction Branching

- At the price of solving the mentioned bilevel program one can be very sophisticated in selecting the disjunction. [Lodi, Ralphs, Rossi, Smriglio, 2009]

- Consider, for the sake of simplicity, a binary integer program

$$\min\{c^T x \mid x \in \mathcal{P} \cap \mathbb{B}^n\},$$

where $c \in \mathbb{Q}^n$ is the objective function and $\mathcal{P}$ is a polyhedron.

- We are interested in a set $S = \{i_1, \dots, i_{|S|}\} \subseteq I^n = \{1, \dots, n\}$, to impose the (clearly valid) multi-variable disjunction

$$x_{i_1} = 1 \vee (x_{i_2} = 1 \wedge x_{i_1} = 0) \vee \dots \vee (x_{i_{|S|}} = 1 \wedge x_{i_1} = 0 \wedge \dots \wedge x_{i_{|S|-1}} = 0) \vee \sum_{i \in S} x_i = 0.$$

- In particular we want sets S playing a fundamental role for improving the current incumbent solution value, say $\bar{z}$.

1: Interdiction Branching (cont.d)

- Such a goal can be achieved by solving the Interdiction Branching Problem (IBP) (at node a):

$$\min \sum_{i \in I^n} y_i \tag{3}$$

$$\text{s.t.} \sum_{i \in I^n} c_i x_i \geq \bar{z} \tag{4}$$

$$y \in \{0,1\}^n \tag{5}$$

$$x \in \arg\min_x c^T x$$

1: Interdiction Branching (cont.d)

- Such a goal can be achieved by solving the Interdiction Branching Problem (IBP) (at node a):

$$\min \sum_{i \in I^n} y_i \tag{3}$$

$$\text{s.t.} \sum_{i \in I^n} c_i x_i \geq \bar{z} \tag{4}$$

$$y \in \{0, 1\}^n \tag{5}$$

$$x \in \arg\min_x c^T x \tag{6}$$

$$\text{s.t. } x_i + y_i \leq 1, \; i \in I^n \tag{7}$$

$$x \in \mathcal{F}(a). \tag{8}$$

where $\mathcal{F}(a)$ denotes the set of feasible solutions of the original problem at node a of the tree.

1: Interdiction Branching (cont.d)

- Such a goal can be achieved by solving the Interdiction Branching Problem (IBP) (at node a):

$$\min \sum_{i \in I^n} y_i \tag{3}$$

$$\text{s.t.} \sum_{i \in I^n} c_i x_i \geq \bar{z} \tag{4}$$

$$y \in \{0,1\}^n \tag{5}$$

$$x \in \arg\min_x c^T x \tag{6}$$

$$\text{s.t. } x_i + y_i \leq 1, \; i \in I^n \tag{7}$$

$$x \in \mathcal{F}(a). \tag{8}$$

where $\mathcal{F}(a)$ denotes the set of feasible solutions of the original problem at node a of the tree.

- In other words,

(i) the last term of the branching disjunction does not have to be explored because it contains NO improving solution, while,

(ii) by the minimality of (3), it follows that any other child node contains at least one improving solution.

1: Interdiction Branching (cont.d)

- Of course, the property (ii) is very strong and, because we are not expecting any "free lunch", solving IBPs MUST be very difficult.

1: Interdiction Branching (cont.d)

- Of course, the property (ii) is very strong and, because we are not expecting any "free lunch", solving IBPs MUST be very difficult.
- However, if carefully treated, most of the good features of the method hold also if the IBP is not solved to optimality.

1: Interdiction Branching (cont.d)

- Of course, the property (ii) is very strong and, because we are not expecting any "free lunch", solving IBPs MUST be very difficult.
- However, if carefully treated, most of the good features of the method hold also if the IBP is not solved to optimality.

- Interdiction branching takes into account both the current incumbent value and the bound provided by the LP relaxation and can thus be seen as targeting improvements in both the upper and lower bounds.
- As in the traditional branching on variables, child subproblems are generated by imposing variable bounds, without introducing additional constraints.
- In all but one of the $|S|$ children, at least two variables are fixed, often yielding a remarkable improvement in the bound provided by the LP relaxation.

1: Interdiction Branching (cont.d)

- Of course, the property (ii) is very strong and, because we are not expecting any "free lunch", solving IBPs MUST be very difficult.
- However, if carefully treated, most of the good features of the method hold also if the IBP is not solved to optimality.

- Interdiction branching takes into account both the current incumbent value and the bound provided by the LP relaxation and can thus be seen as targeting improvements in both the upper and lower bounds.
- As in the traditional branching on variables, child subproblems are generated by imposing variable bounds, without introducing additional constraints.
- In all but one of the $|S|$ children, at least two variables are fixed, often yielding a remarkable improvement in the bound provided by the LP relaxation.

- Extensive computational experiments on difficult 0-1 knapsack instances have shown a very favorable behavior wrt traditional branching on variables although this has to be confirmed within state-of-the-art solvers.

2: Traditional Cut Genetation

- Current MIP cutting plane technology is based on the following two-step heuristic:

2: Traditional Cut Genetation

- Current MIP cutting plane technology is based on the following two-step heuristic:

  1. heuristically aggregate the entire MIP into a mixed-integer set of one single (!) row

  2. apply the cut separation procedure to such a mixed-integer set
     (Chvátal-Gomory, Gomory Mixed-Integer, MIR, $\{0, \frac{1}{2}\}$, Cover, . . .).

2: Traditional Cut Genetation

- Current MIP cutting plane technology is based on the following two-step heuristic:

  1. heuristically aggregate the entire MIP into a mixed-integer set of one single (!) row
  2. apply the cut separation procedure to such a mixed-integer set (Chvátal-Gomory, Gomory Mixed-Integer, MIR, $\{0, \frac{1}{2}\}$, Cover, . . .).

- In the most natural setting, the aggregation of step 1 above is done by using a single row of the simplex tableau as mixed-integer set.

2: Traditional Cut Genetation

- Current MIP cutting plane technology is based on the following two-step heuristic:

  1. heuristically aggregate the entire MIP into a mixed-integer set of one single (!) row

  2. apply the cut separation procedure to such a mixed-integer set (Chvátal-Gomory, Gomory Mixed-Integer, MIR, $\{0, \frac{1}{2}\}$, Cover, . . .).

- In the most natural setting, the aggregation of step 1 above is done by using a single row of the simplex tableau as mixed-integer set.

- Very recently, since 2007, a new line of research (strongly connected to a very old one) has been followed by considering multiple rows of the simplex tableau at the same time.

2: Cuts from Multiple Rows of the Simplex Tableau

- We consider a mixed-integer set of the form

$$S = \{x \in \mathbb{R}^n \ : \ Ax = b, x \geq 0, x_j \in \mathbb{Z} \ \forall j \in \mathcal{I}\}$$

with $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$.

2: Cuts from Multiple Rows of the Simplex Tableau

- We consider a mixed-integer set of the form

$$S = \{x \in \mathbb{R}^n \ : \ Ax = b, x \geq 0, x_j \in \mathbb{Z} \ \forall j \in \mathcal{I}\}$$

with $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$.

- Given a basis $B \subset \{1, \ldots, n\}$ corresponding to a vertex x^* of the continuous relaxation of S, the set S can be rewritten as

$$\begin{array}{rcl} x_B & = & x^*_B + \sum_{j \in N} r^j x_j, \\ x & \geq & 0, \\ x_j & \in & \mathbb{Z}, \ j \in \mathcal{I}, \end{array} \tag{9}$$

where N denotes the set of nonbasic variables.

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- A first relaxation of S can be obtained by dropping the nonnegativity restrictions on all the basic variables and considering a subset Q (with $|Q| = q$) of rows of (9) associated with basic integer-constrained variables, thus getting

$$\begin{array}{llll} (S_Q) & x_i & = & f_i + \sum_{j \in N} r_i^j x_j, \;\; i \in Q \\ & x_j & \geq & 0, \;\; j \in N \\ & x_j & \in & \mathbb{Z}, \;\; j \in \mathcal{I} \end{array} \qquad (10)$$

with $f_i = x_i^* - \lfloor x_i^* \rfloor$ for any $i \in Q$ and $f_i > 0$ for some $i \in Q = \{1, \ldots, q\}$.

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- A first relaxation of S can be obtained by dropping the nonnegativity restrictions on all the basic variables and considering a subset Q (with $|Q| = q$) of rows of (9) associated with basic integer-constrained variables, thus getting

$$
\begin{array}{llll}
(S_Q) & x_i & = & f_i + \sum_{j \in N} r_i^j x_j, \;\; i \in Q \\
 & x_j & \geq & 0, \;\; j \in N \\
 & x_j & \in & \mathbb{Z}, \;\; j \in \mathcal{I}
\end{array}
\tag{10}
$$

  with $f_i = x_i^* - \lfloor x_i^* \rfloor$ for any $i \in Q$ and $f_i > 0$ for some $i \in Q = \{1, \ldots, q\}$.

- A key step is further relaxing set S_Q by dropping all the integrality requirements on the nonbasic variables, thus getting a system of the form

$$
\begin{array}{lll}
x & = & f + \sum_{j=1}^{k} r^j s_j, \\
x & \in & \mathbb{Z}^q, \\
s & \in & \mathbb{R}^k_+,
\end{array}
\tag{11}
$$

  where all the continuous variables have been renamed as s and $|N| = k$.

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- A first relaxation of S can be obtained by dropping the nonnegativity restrictions on all the basic variables and considering a subset Q (with $|Q| = q$) of rows of (9) associated with basic integer-constrained variables, thus getting

$$(S_Q) \quad \begin{array}{rcl} x_i & = & f_i + \sum_{j \in N} r_i^j x_j, \;\; i \in Q \\ x_j & \geq & 0, \;\; j \in N \\ x_j & \in & \mathbb{Z}, \;\; j \in \mathcal{I} \end{array} \tag{10}$$

  with $f_i = x_i^* - \lfloor x_i^* \rfloor$ for any $i \in Q$ and $f_i > 0$ for some $i \in Q = \{1, \ldots, q\}$.

- A key step is further relaxing set S_Q by dropping all the integrality requirements on the nonbasic variables, thus getting a system of the form

$$\begin{array}{rcl} x & = & f + \sum_{j=1}^{k} r^j s_j, \\ x & \in & \mathbb{Z}^q, \\ s & \in & \mathbb{R}_+^k, \end{array} \tag{11}$$

  where all the continuous variables have been renamed as s and $|N| = k$.

- Denote as $R_f(r^1, \ldots, r^k)$ the convex hull of all vectors $s \in \mathbb{R}^k$ for which there exists $x \in \mathbb{R}^q$ such that (x, s) satisfies (11).

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- Borozan & Cornuéjols considered relaxing the k-dimensional space of variables $s = (s_1, \ldots, s_k)$ to an infinite-dimensional space, where the variables s_r are defined for any $r \in \mathbb{Q}^q$.

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- Borozan & Cornuéjols considered relaxing the k-dimensional space of variables $s = (s_1, \dots, s_k)$ to an infinite-dimensional space, where the variables s_r are defined for any $r \in \mathbb{Q}^q$.
- This is the semi-infinite relaxation and is strongly related to Gomory and Johnson's infinite group problem. [Gomory & Johnson, 1972]

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- Borozan & Cornuéjols considered relaxing the k-dimensional space of variables $s = (s_1, \ldots, s_k)$ to an infinite-dimensional space, where the variables s_r are defined for any $r \in \mathbb{Q}^q$.

- This is the semi-infinite relaxation and is strongly related to Gomory and Johnson's infinite group problem. [Gomory & Johnson, 1972]

- In addition, Borozan & Cornuéjols proved that any valid inequality for R_f can be written as

$$\sum_{r \in \mathbb{Q}^q} \psi(r) s_r \geq 1 \tag{12}$$

where $\psi : \mathbb{Q}^q \longrightarrow \mathbb{Q} \cup \{+\infty\}$ is said to be a valid function if the corresponding inequality (12) is valid for R_f.

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- Borozan & Cornuéjols considered relaxing the k-dimensional space of variables $s = (s_1, \ldots, s_k)$ to an infinite-dimensional space, where the variables s_r are defined for any $r \in \mathbb{Q}^q$.

- This is the semi-infinite relaxation and is strongly related to Gomory and Johnson's infinite group problem. [Gomory & Johnson, 1972]

- In addition, Borozan & Cornuéjols proved that any valid inequality for R_f can be written as

$$\sum_{r \in \mathbb{Q}^q} \psi(r) s_r \geq 1 \tag{12}$$

where $\psi : \mathbb{Q}^q \longrightarrow \mathbb{Q} \cup \{+\infty\}$ is said to be a valid function if the corresponding inequality (12) is valid for R_f.

- Finally, Borozan & Cornuéjols provided a strong correspondence between minimal valid inequalities and maximal lattice-free convex sets.

2: Cuts from Multiple Rows of the Simplex Tableau (cont.d)

- Borozan & Cornuéjols considered relaxing the k-dimensional space of variables $s = (s_1, \ldots, s_k)$ to an infinite-dimensional space, where the variables s_r are defined for any $r \in \mathbb{Q}^q$.

- This is the semi-infinite relaxation and is strongly related to Gomory and Johnson's infinite group problem. [Gomory & Johnson, 1972]

- In addition, Borozan & Cornuéjols proved that any valid inequality for R_f can be written as

$$\sum_{r \in \mathbb{Q}^q} \psi(r) s_r \geq 1 \tag{12}$$

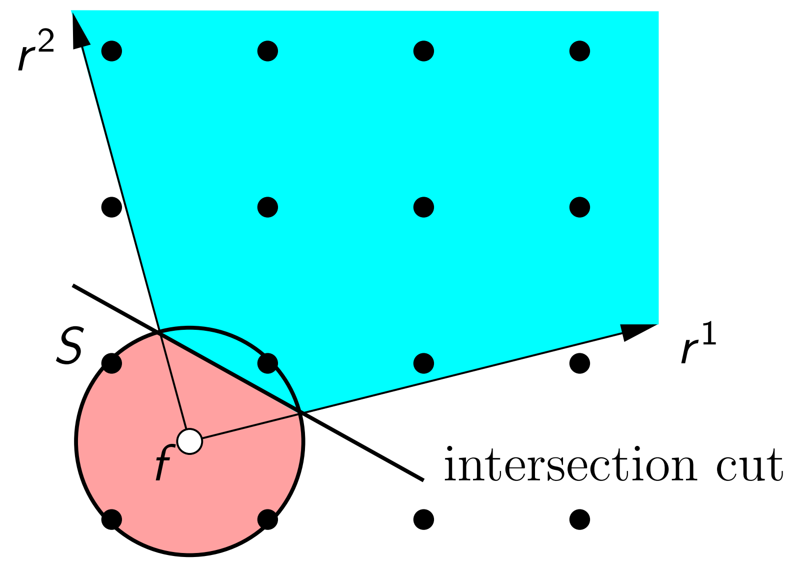
where $\psi : \mathbb{Q}^q \longrightarrow \mathbb{Q} \cup \{+\infty\}$ is said to be a valid function if the corresponding inequality (12) is valid for R_f.

- Finally, Borozan & Cornuéjols provided a strong correspondence between minimal valid inequalities and maximal lattice-free convex sets.

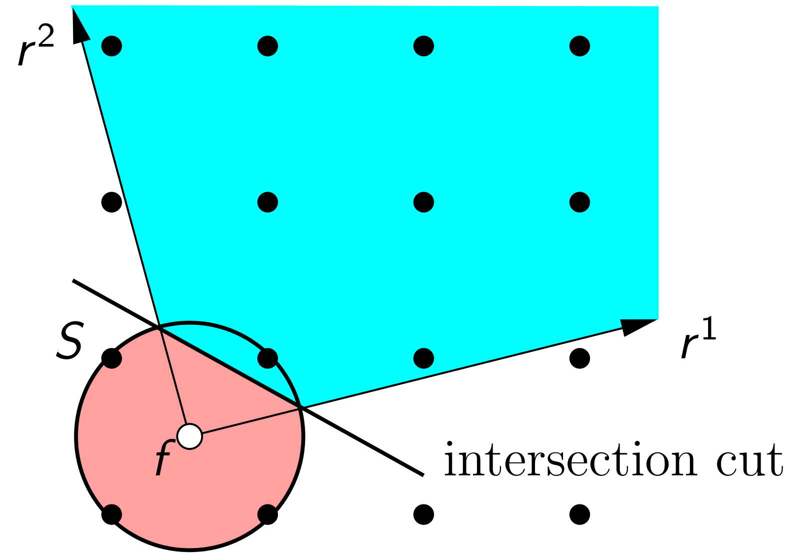
- Zambelli proved that one does not have to worry about lattice-free convex sets with f on their boundary.

2: Cuts from TWO Rows of the Simplex Tableau

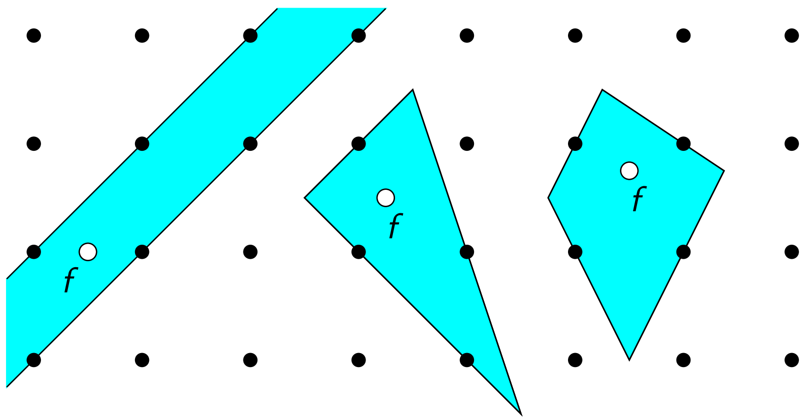
Andersen, Louveaux, Weismantel & Wolsey have shown that all the facets of $R_f(r^1, \ldots, r^k)$ are intersection cuts arising from two-dimensional lattice-free convex sets. [Balas, 1971]



2: Cuts from TWO Rows of the Simplex Tableau



Andersen, Louveaux, Weismantel & Wolsey have shown that all the facets of $R_f(r^1, \ldots, r^k)$ are intersection cuts arising from two-dimensional lattice-free convex sets. [Balas, 1971]



Cornuéjols & Margot proved that only three types of maximal lattice-free convex sets are sufficient, namely, splits, triangles and quadrilaterals.

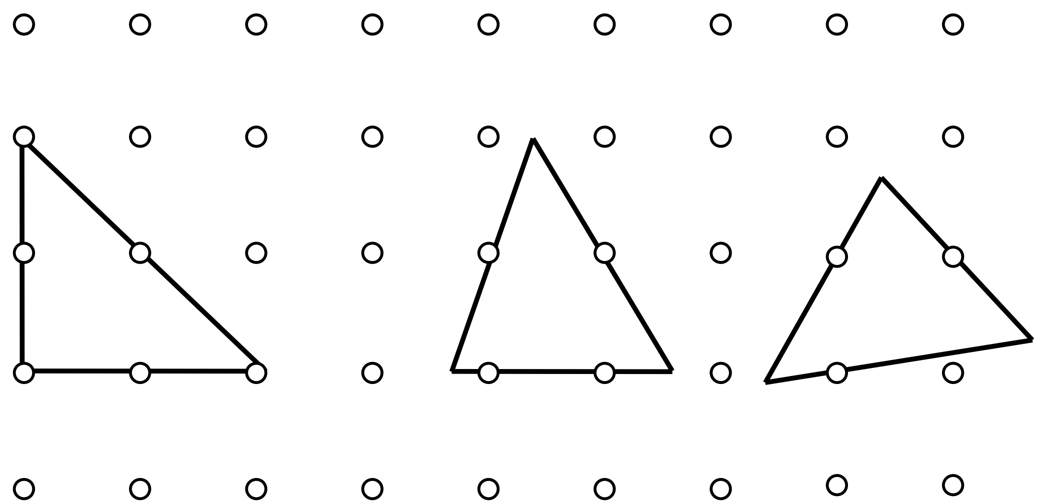
2: TWO Row Cuts, Possible Triangles

- Dey & Wolsey characterized the maximal lattice-free triangles. Namely,

2: TWO Row Cuts, Possible Triangles

- Dey & Wolsey characterized the maximal lattice-free triangles. Namely,
  If Π is a maximal lattice-free triangle in $\mathbb{R}^2$, then exactly one of the following is true:

1. All the vertices are integral and each side contains one integral point in its relative interior.
2. One side of Π contains more than one integral point in its relative interior.
3. The vertices are non-integral and each side contains one integral point in its relative interior.



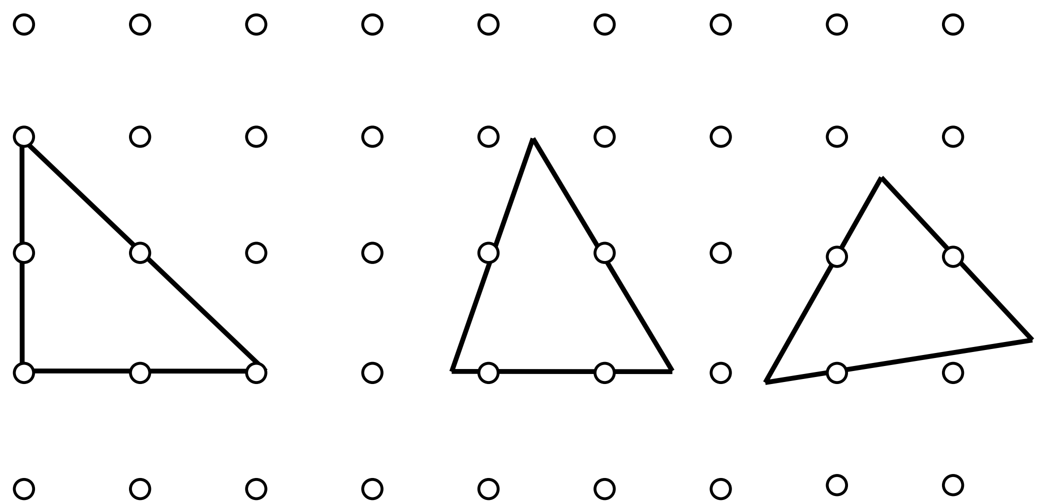
2: TWO Row Cuts, Possible Triangles

- Dey & Wolsey characterized the maximal lattice-free triangles. Namely,
  If Π is a maximal lattice-free triangle in $\mathbb{R}^2$, then exactly one of the following is true:

  1. All the vertices are integral and each side contains one integral point in its relative interior.

  2. One side of Π contains more than one integral point in its relative interior.

  3. The vertices are non-integral and each side contains one integral point in its relative interior.



- For the triangles of type 1 and 2 above they also defined the best possible way of lifting the coefficients of the nonbasic integer variables.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals.
  First, they proved that triangle and quadrilateral closures are at least as strong as the split one.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals. First, they proved that triangle and quadrilateral closures are at least as strong as the split one. Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals. First, they proved that triangle and quadrilateral closures are at least as strong as the split one. Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

- At this point, there was a lot of wisdom on the strength of these cuts because, in practice, the split closure is already very tight.
- However, evaluating multi-row cuts from a computational viewpoint is somehow more involved.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals. First, they proved that triangle and quadrilateral closures are at least as strong as the split one. Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

- At this point, there was a lot of wisdom on the strength of these cuts because, in practice, the split closure is already very tight.
- However, evaluating multi-row cuts from a computational viewpoint is somehow more involved.
  - The first computational results involving the separation of cuts from q rows has been reported by Espinoza on a large set of MIPLIB instances.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals. First, they proved that triangle and quadrilateral closures are at least as strong as the split one. Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

- At this point, there was a lot of wisdom on the strength of these cuts because, in practice, the split closure is already very tight.
- However, evaluating multi-row cuts from a computational viewpoint is somehow more involved.
  - The first computational results involving the separation of cuts from q rows has been reported by Espinoza on a large set of MIPLIB instances.
  - Basu, Bonami, Cornuéjols & Margot studied special two-row cuts with either f_1 or f_2 is 0.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals. First, they proved that triangle and quadrilateral closures are at least as strong as the split one. Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

- At this point, there was a lot of wisdom on the strength of these cuts because, in practice, the split closure is already very tight.
- However, evaluating multi-row cuts from a computational viewpoint is somehow more involved.
  - The first computational results involving the separation of cuts from q rows has been reported by Espinoza on a large set of MIPLIB instances.
  - Basu, Bonami, Cornuéjols & Margot studied special two-row cuts with either f_1 or f_2 is 0.
  - Dey, Lodi, Tramontani & Wolsey extensively tested the impact of two-row cuts from triangles of type 2 on multiple knapsack instances.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals. First, they proved that triangle and quadrilateral closures are at least as strong as the split one. Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

- At this point, there was a lot of wisdom on the strength of these cuts because, in practice, the split closure is already very tight.
- However, evaluating multi-row cuts from a computational viewpoint is somehow more involved.
  - The first computational results involving the separation of cuts from q rows has been reported by Espinoza on a large set of MIPLIB instances.
  - Basu, Bonami, Cornuéjols & Margot studied special two-row cuts with either f_1 or f_2 is 0.
  - Dey, Lodi, Tramontani & Wolsey extensively tested the impact of two-row cuts from triangles of type 2 on multiple knapsack instances.

- Overall, the results are mixed.

2: Impact of Multiple Row Cuts

- On the theoretical side, Basu, Bonami, Cornuéjols & Margot compared the strength of the closures of the different lattice-free convex sets, i.e., splits, triangles and quadrilaterals.
  First, they proved that triangle and quadrilateral closures are at least as strong as the split one.
  Then, they showed how triangle and quadrilateral closures always provide a good approximation of $R_f(r^1, \ldots, r^k)$ in a well defined sense (i.e., they always close at least half of the integrality gap), while the approximation provided by the split closure can be arbitrarily bad.

- At this point, there was a lot of wisdom on the strength of these cuts because, in practice, the split closure is already very tight.
- However, evaluating multi-row cuts from a computational viewpoint is somehow more involved.
  - The first computational results involving the separation of cuts from q rows has been reported by Espinoza on a large set of MIPLIB instances.
  - Basu, Bonami, Cornuéjols & Margot studied special two-row cuts with either f_1 or f_2 is 0.
  - Dey, Lodi, Tramontani & Wolsey extensively tested the impact of two-row cuts from triangles of type 2 on multiple knapsack instances.

- Overall, the results are mixed.
  Before really asserting their effectiveness one has to find a clever way of doing cut selection which also includes understanding their relationship with Mixed-Integer Gomory cuts.