Recent Developments in Cuts for MILP
3. Empirical tests

François Margot

Carnegie Mellon University

March 2013

Mixed-Integer Linear Program (MILP)

$$
\begin{array}{lll}
\max & c^T x & \\
\text{s.t.} & Ax = b & \\
& x_i \in \mathbb{Z} & \text{for } i \in I \\
& x_i \geq 0 & \text{for } i \in [n]
\end{array}
$$

2-dimensional Relaxed Corner Polyhedron:

$$
(RCP): \qquad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} + \sum_{j \in N} \begin{pmatrix} r_1^j \\ r_2^j \end{pmatrix} x_j
$$

$$
x_j \geq 0 \text{ for all } j \in N
$$

$$
x_1, x_2 \in \mathbb{Z}
$$

$\Gamma = \{r^j \mid j \in N\}$

$R(f, \Gamma)$ = convex hull of (RCP)

(LRCP): Linear relaxation of (RCP)

Empirical tests

2-row cuts

- Type 1 and Type 2 triangles

  [Basu, Bonami, Cornuéjols, Margot 2011 [BBCM]]

- Type 2 triangles

  [Dey, Lodi, Tramontani, Wolsey 2012 [DLTW]]

- Parametric octahedron

  [Balas, Qualizza 2012 [BQ]]

- Using the polar

  [Louveaux, Poirrier 2012 [LP]]

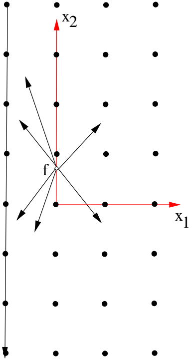
5-row 10-row or 15-row cuts:

- Generalized Type 1 triangles, octahedron

  [Espinoza 2010 [E]]

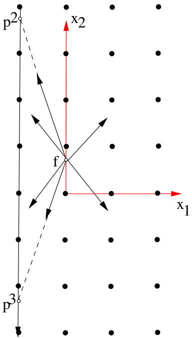
Separation algorithm [BBCM]

- Assume that $f = (0, f_2)$ with $0 < f_2 < 1$



Separation algorithm [BBCM]

- Assume that $f = (0, f_2)$ with $0 < f_2 < 1$
- $p^2 = (-1, p_2^2)$: intersection of ray with $x_1 = -1$ with largest x_2-coord
- $p^3 = (-1, p_2^3)$: intersection of ray with $x_1 = -1$ with smallest x_2-coord



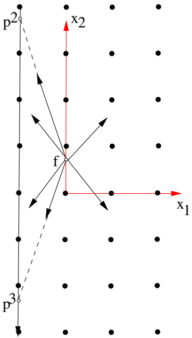
Separation algorithm [BBCM]

- Assume that $f = (0, f_2)$ with $0 < f_2 < 1$

- $p^2 = (-1, p^2_2)$: intersection of ray with $x_1 = -1$ with largest x_2-coord

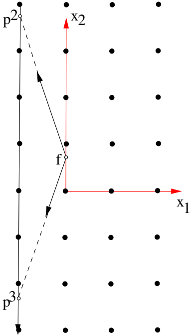
- $p^3 = (-1, p^3_2)$: intersection of ray with $x_1 = -1$ with smallest x_2-coord

- If $p^2 = p^3$ or if they do not exist, stop.



Separation algorithm (cont.)

Otherwise, $p^2 \neq p^3$ well defined



Separation algorithm (cont.)

Otherwise, $p^2 \neq p^3$ well defined

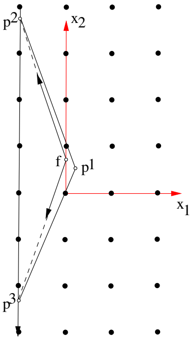
- If at least two integer points in $\mathrm{int}(p^2 p^3)$:

  L^2 line through p^2 and $(0,1)$

  L^3 line through p^3 and $(0,0)$

  $p^1 = L^2 \cap L^3$

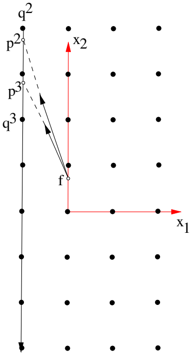
  Type 2 triangle $p^1 p^2 p^3$



Separation algorithm (cont.)

- If exactly one integer point in $\mathrm{int}(p^2 p^3)$:

$$q^2 = (-1, \lceil p_2^2 \rceil)$$
$$q^3 = (-1, \lfloor p_2^3 \rfloor)$$

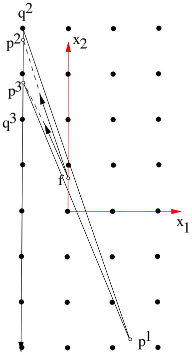


Separation algorithm (cont.)

- If exactly one integer point in $\text{int}(p^2p^3)$:

  $q^2 = (-1, \lceil p_2^2 \rceil)$
  $q^3 = (-1, \lfloor p_2^3 \rfloor)$

  - If $\lceil p_2^2 \rceil - p_2^2 \leq p_2^3 - \lfloor p_2^3 \rfloor$
    L^2 line through q^2 and $(0, 1)$
    L^3 line through p^3 and $(0, 0)$
    $p^1 = L^2 \cap L^3$
    Type 2 triangle $p^1q^2p^3$



Separation algorithm (cont.)

- If exactly one integer point in $\mathrm{int}(p^2p^3)$:

  $q^2 = (-1, \lceil p^2_2 \rceil)$
  $q^3 = (-1, \lfloor p^3_2 \rfloor)$

  - If $\lceil p^2_2 \rceil - p^2_2 \leq p^3_2 - \lfloor p^3_2 \rfloor$

    L^2 line through q^2 and $(0, 1)$

    L^3 line through p^3 and $(0, 0)$

    $p^1 = L^2 \cap L^3$

    Type 2 triangle $p^1q^2p^3$

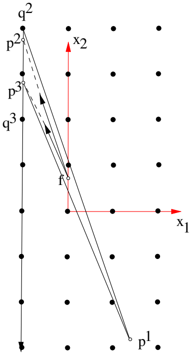
  - Otherwise

    L^2 line through p^2 and $(0, 1)$

    L^3 line through q^3 and $(0, 0)$

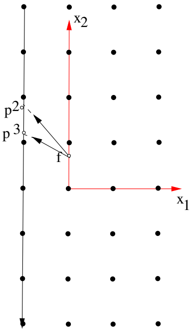
    $p^1 = L^2 \cap L^3$

    Type 2 triangle $p^1p^2q^3$



Separation algorithm (cont.)

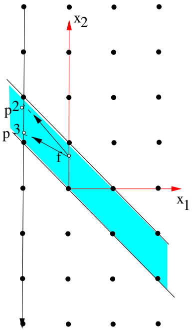
- If no integer point in $\mathrm{int}(p^2 p^3)$:



Separation algorithm (cont.)

- If no integer point in $\mathrm{int}(p^2 p^3)$:

  Split with sides through $(0,1)$ and $(0,0)$ and containing $p^2 p^3$



Comparing cut generators

Algorithms:

- `G`: Gomory cut generator
- `G-2Rounds`: Gomory cut generator, two rounds
- `G + Allpairs`: `G` + triangles and splits using above heuristic for all pairs of rows $x_i \in \mathbb{Z}$, x_j fractional
- `G + Deepest`: `G` + deepest triangle or split using above heuristic for each pairs of rows $x_i \in \mathbb{Z}$, x_j fractional

Empirical Testing of Algorithms

[Hooker 1994 [H]], [Hooker 1995 [H95]]
[McGeoch 2001 [MG]], [McGeoch 2012 [MG12]]

- Run experiments
- Formulate a hypothesis about an algorithm
- Use controlled experiments to test the validity of the hypothesis using statistical analysis tools

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
  - Choice of problems

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
  - Choice of problems
  - Use other cut generators: Tough to see effect of one cut generator

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
    - Choice of problems
    - Use other cut generators: Tough to see effect of one cut generator
    - Use heuristics: May hide precision problem

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
  - Choice of problems
  - Use other cut generators: Tough to see effect of one cut generator
  - Use heuristics: May hide precision problem
  - Do not use heuristics or other cut generators: Longer solution times

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
  - Choice of problems
  - Use other cut generators: Tough to see effect of one cut generator
  - Use heuristics: May hide precision problem
  - Do not use heuristics or other cut generators: Longer solution times
  - Interpretation of different “optimal” solutions found?

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
  - Choice of problems
  - Use other cut generators: Tough to see effect of one cut generator
  - Use heuristics: May hide precision problem
  - Do not use heuristics or other cut generators: Longer solution times
  - Interpretation of different “optimal” solutions found?
  - Small differences on most problems

Standard procedure I

- Benchmark a Branch-and-Cut code on a collection of problems
  - Choice of problems
  - Use other cut generators: Tough to see effect of one cut generator
  - Use heuristics: May hide precision problem
  - Do not use heuristics or other cut generators: Longer solution times
  - Interpretation of different “optimal” solutions found?
  - Small differences on most problems
  - Statistical significance of the results?

Standard procedure II

- Benchmark lower bound at the root on a collection of problems

Standard procedure II

- Benchmark lower bound at the root on a collection of problems
  - Choice of problems

Standard procedure II

- Benchmark lower bound at the root on a collection of problems
    - Choice of problems
    - Strength at root not always a good indicator

Standard procedure II

- Benchmark lower bound at the root on a collection of problems
  - Choice of problems
  - Strength at root not always a good indicator
  - Trade off between speed and lower bound

Standard procedure II

- Benchmark lower bound at the root on a collection of problems
  - Choice of problems
  - Strength at root not always a good indicator
  - Trade off between speed and lower bound
  - Inappropriate to gauge precision

Standard procedure II

- Benchmark lower bound at the root on a collection of problems
  - Choice of problems
  - Strength at root not always a good indicator
  - Trade off between speed and lower bound
  - Inappropriate to gauge precision

Two better empirical testing setups:

- Random Diving Towards Optimal Solution [Margot 2009 [M]]
- Dive-and-cut [Cornuéjols, Margot, Nannicini 2013 [CMN]]

Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

- Record optimal solution (x^P, y^P)

  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached

Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

- Record optimal solution (x^P, y^P)

  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached

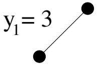
-

Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

- Record optimal solution (x^P, y^P)

  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached

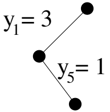


Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

- Record optimal solution (x^P, y^P)

  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached

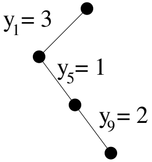


Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

- Record optimal solution (x^P, y^P)

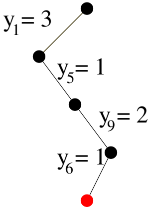
  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached



Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

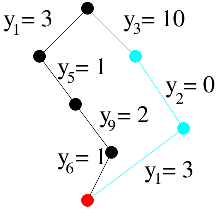
- Record optimal solution (x^P, y^P)
- Repeat
  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached



Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

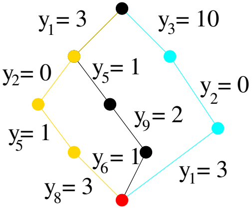
- Record optimal solution (x^P, y^P)
- Repeat
    - Start with original LP formulation
    - Repeat
        - Generate and apply k rounds of cuts
        - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
        - Set $y_j = y_j^P$
        - Check if (x^P, y^P) is still feasible or not (with some tolerance)
        - Exit if time limit is reached



Random Diving Towards Optimal Solution [M]

For each problem P in a given collection:

- Record optimal solution (x^P, y^P)
- Repeat
  - Start with original LP formulation
  - Repeat
    - Generate and apply k rounds of cuts
    - Select randomly an integer variable y_j with fractional value in LP solution (Exit if none)
    - Set $y_j = y_j^P$
    - Check if (x^P, y^P) is still feasible or not (with some tolerance)
    - Exit if time limit is reached



Dive-and-Cut [CMN]

Instance I with known feasible solutions S

Idea:

- Select randomly a node N of a Branch-and-Bound solving I
- Generate cuts
- test feasibility using solutions in S valid for N

Dive-and-Cut [CMN]

Instance I with known feasible solutions S

Idea:

- Select randomly a node N of a Branch-and-Bound solving I
- Generate cuts
- test feasibility using solutions in S valid for N

Dive-and-Cut:

- Select randomly $s \in S$
- Select randomly a fraction $t \in [0, 0.8]$
- Fix randomly integer variables to their value in s until fraction t of the initial gap is closed
- Generate ρ rounds of cuts
- Use all solutions in S still valid for N to test validity

Gap closed at different depths

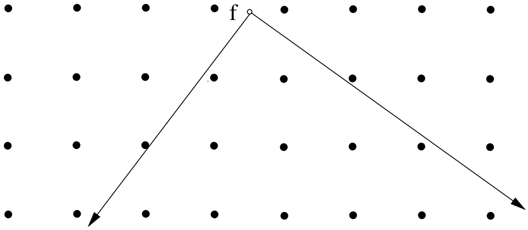
MIPLIB3_C_V2 using Random Diving

	Depth 0 (root node)	Depth 4	Depth 8	Depth 12
G	28.33	57.41	68.15	75.23
G+Allpairs	29.11	58.13	68.43	74.81
G+Deepest	28.80	58.09	68.59	75.40
G-2Rounds	36.66	59.41	68.75	75.47

- G vs. G+Allpairs: marginal improvement
- G+Allpairs vs. G+Deepest: G+Deepest better; significant at depth 12
- G-2Rounds vs. G+Deepest: G-2Rounds better at depth 0 and 4 (signif.); difference at depth 8 and 12 not significant.

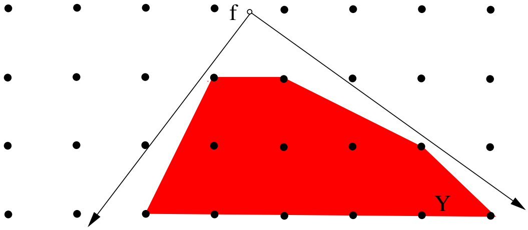
Use Quade test for statistical tests

Separation for Type 2 triangles [DLTW]



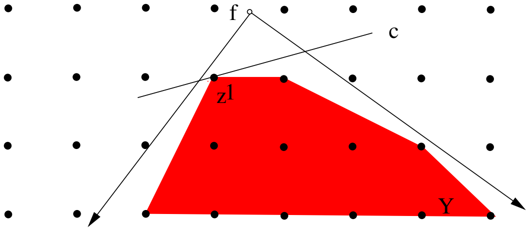
Choose two rays r^1, r^2

Separation for Type 2 triangles [DLTW]



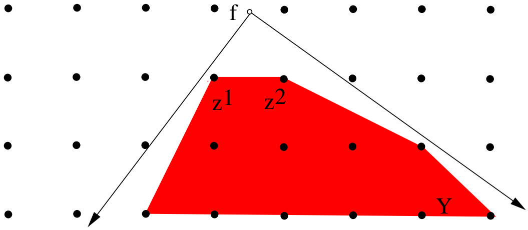
Use integer hull Y of points $z = \lambda_1 r^1 + \lambda_2 r^2$
with $z \in \mathbb{Z}^2$ and $\lambda_1, \lambda_2 \geq 0$

Separation for Type 2 triangles [DLTW]



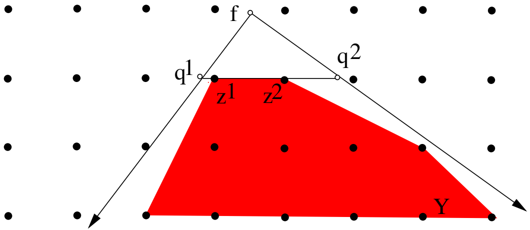
Optimize bisector c of r^1 and r^2 over $Y \to z^1$

Separation for Type 2 triangles [DLTW]



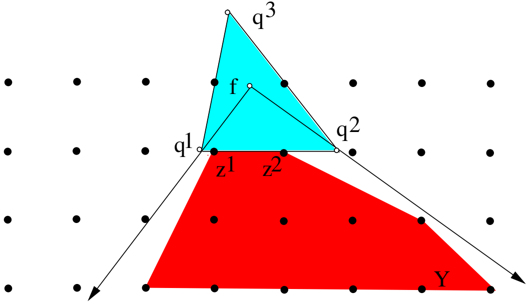
Compute a neighbor z^2 of z^1 on boundary of Y
(iterative ILP)

Separation for Type 2 triangles [DLTW]



Side of triangle through $z^1 z^2$, endpoints q^1, q^2 on r^1, r^2

Separation for Type 2 triangles [DLTW]



Complete to a Type 2 triangle containing f

Gap closed on random 2-row instances

Density	G	S	T	GS	GT	GST
100%	75.10 (0.16)	76.39 (0.17)	97.99 (0.05)	92.69 (0.07)	98.01 (0.05)	98.94 (0.03)
80%	74.79 (0.18)	66.82 (0.23)	94.02 (0.09)	92.32 (0.08)	96.66 (0.07)	98.43 (0.04)
60%	80.06 (0.16)	56.79 (0.27)	91.38 (0.14)	90.29 (0.10)	97.19 (0.06)	97.70 (0.05)

- `G`: Gomory cut generator
- `S`: Non-simple split cut from 2-rows (coeff. of split in $[-3, 3]$)
- `T`: For each pair of rays, generate one Type 2 triangle using the heuristic (use only rows with fractional basic variable that should be integer)

Findings on random instances

5-row instances:

- All cut families are weaker; 2-row cuts are stronger than 1-row cuts
- `T` vs. `S`: If continuous variables are important, `T` better than `S` and vice-versa
- When many integer non-basic variables, all cuts become weaker; need other relaxations

Adding bounds on variables:

- Does not affect much comparisons 2-row vs. 1-row
- Does not affect much comparisons `T` vs. `S`

Findings on random instances (cont.)

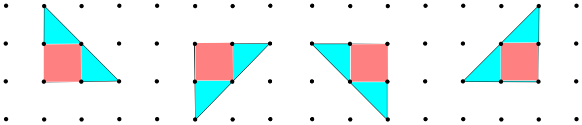
m-row instances, various density:

- 2-row vs. 1-row:
  - Combining `G` and `T` much better than `G`
  - `T` more important for very dense instances
- `T` vs. `S`:
  - `T` stronger than `GS` only for very dense instances
  - `G` stronger than `T` on sparse instances

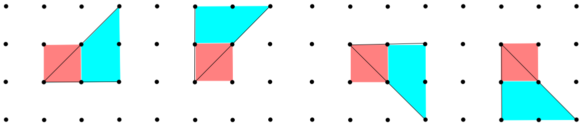
Parametric octahedron [BQ]

Assume that $0 < f_i < 1$ for $i = 1, 2$

Use the four Type 1 triangles containing unit square U:



Use eight cones whose vertex is a vertex of U and one ray horizontal or vertical and the other diagonal:



Parametric octahedron (cont.)

MIPLIB3_C_V2

	Gap closed	# cuts
G	19.49	19.54
ParamOct	29.06	3903

- Weak Gomory generator (28.33 with stronger generator)
- Large number of cuts
- Only average result, no statistical test

Polar [LP]

Polar of $R(f,\Gamma)$:

$$Q = \{\alpha \in \mathbb{R}_+^N \mid \alpha \bar{x}^N \geq 1, \text{for all } \bar{x} \text{ extreme point of } R(f,\Gamma);$$
$$\alpha \bar{t}^N \geq 1, \text{for all } (\bar{r},\bar{t}) \text{ extreme ray of } R(f,\Gamma)\}$$

Separation of $((x^B)^*,(x^N)^*)$ from $R(f,\Gamma)$:

$$\begin{aligned} \min z &= (x^N)^*\alpha \\ \text{s.t.} &\quad \alpha \in Q \end{aligned}$$

Then

- $z \geq 1 \Leftrightarrow ((x^B)^*,(x^N)^*) \in R(f,\Gamma)$
- $z < 1 \Rightarrow \alpha x^N \geq 1$ separates $((x^B)^*,(x^N)^*)$ from $R(f,\Gamma)$

Polar (cont.)

Simplification: For rays in topological order

$C_{i,i+1} = \{x \in \mathbb{R}^2 | \; x = f + r^i x_i + r^{i+1} x_{i+1}, x_i, x_{i+1} \geq 0\}$

$\chi_{i,i+1}$: vertices of $conv(C_{i,i+1} \cap \mathbb{Z}^2)$

For $\bar{x} \in \chi_{i,i+1}$ define $\bar{x} = r^i \bar{s}_i^x + r^{i+1} \bar{s}_{i+1}^x$ For $r^i, r^j, r^k \in \mathbb{R}^2$ define $r^j = \lambda_{i,k}^j r^i + \lambda_{k,i}^j r^k$

$$\begin{aligned}\bar{Q} &= \{\alpha \in \mathbb{R}_+^N \mid \alpha_i \bar{s}_i^x + \alpha_j \bar{s}_j^x \geq 1, \text{for all } \bar{x} \in \chi_{i,i+1}, \\ &\quad \alpha_i \leq \lambda_{i-1,i+1}^i \alpha_{i-1} + \lambda_{i+1,i-1}^i \alpha_{i+1} \text{for all } r^i \in cone(r^{i-1}, r^{i+1})\}\end{aligned}$$

Theorem 1 [LP] For $c \in \mathbb{R}^N$, $c > 0$, $\min\{c\alpha \; | \alpha \in Q\}$ and $\min\{c\alpha \; | \alpha \in \bar{Q}\}$ have the same set of optimal solutions.

Polar (cont.)

Optimizing over $\bar{Q}$:

- Generate a small set $S_{i,i+1} \subseteq \chi_{i,i+1}$ for all i
- Let $S = \cup_i \ S_{i,i+1}$
- $\bar{Q}(S) := \bar{Q}$ with only inequalities from points in S
- Optimize over $\bar{Q}(S) \rightarrow \bar{\alpha}$
- Find $x \in \mathbb{Z}^2$ such that $\bar{\alpha} \notin \bar{Q}(S \cup x)$
- If no such x, stop
- Otherwise, add x to S and iterate

Polar (cont.)

Cut generation loop:

- For $r = 1$ to 5
  1. Optimize over linear relaxation LP
  2. build up to 5,000 2-row models
  3. Generate Gomory cuts and add them to LP
  4. Reoptimize $\rightarrow x^*$
  5. For each 2-row model, try to generate one cut for x^* and add them to LP
  6. If at least one cut found go to 4

	G	# cuts	G+Polar	+cuts	G+Splits	+cuts
MILIB3	29.41	695	36.18	232	34.79	40
MIPLIB2003	31.32	4,465	34.53	600	33.07	465

What is gap closed by `G`*?*

Reference generator `bestgenaway` [CMN]
53 problems from `MIPLIB 3`:

	1 round	2 rounds	3 rounds	4 rounds	5 rounds
[CMN]	23.17	29.57	32.70	34.61	36.03
[LP]					31.02

46 problems from `MIPLIB 3` vs. 46 problems from `MIPLIB_3_C_V2`

	1 round	2 rounds	3 rounds	4 rounds	5 rounds
[CMN]	17.73	22.24	24.57	25.92	27.25
[LP]					21.81
[BBCM]	21.62	26.82			
[BQ]	16.54				