Cutting Planes

Strengthening model formulations on the fly

Timo Berthold
TU Berlin, FICO, MODAL

image source: DALL-E

Agenda

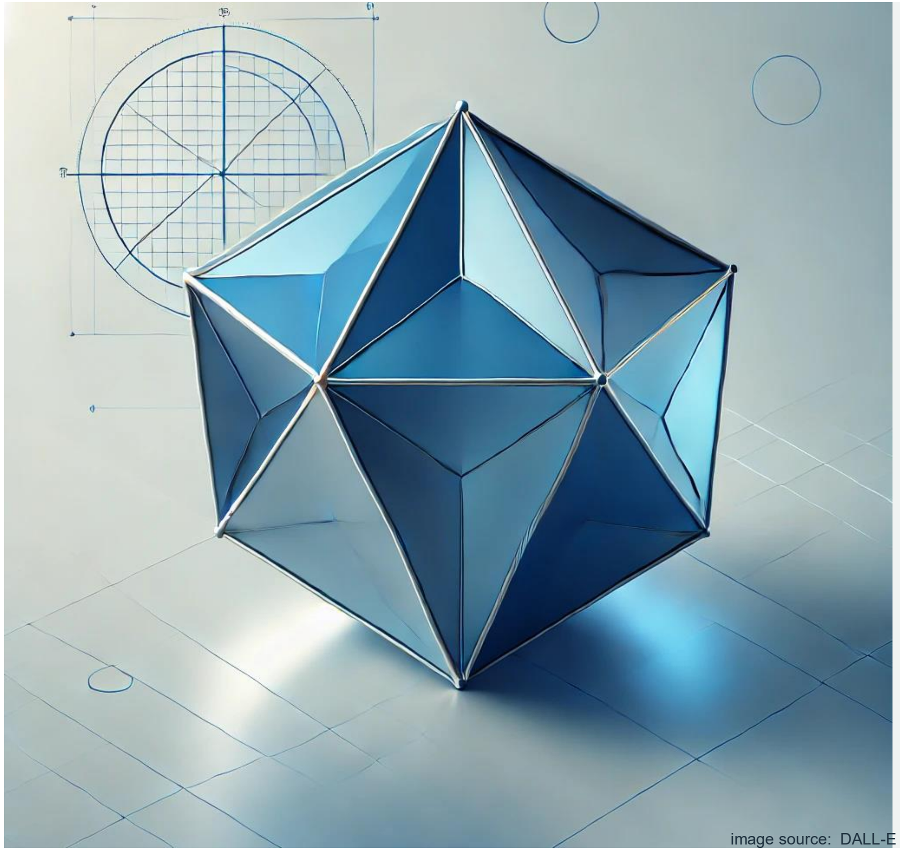


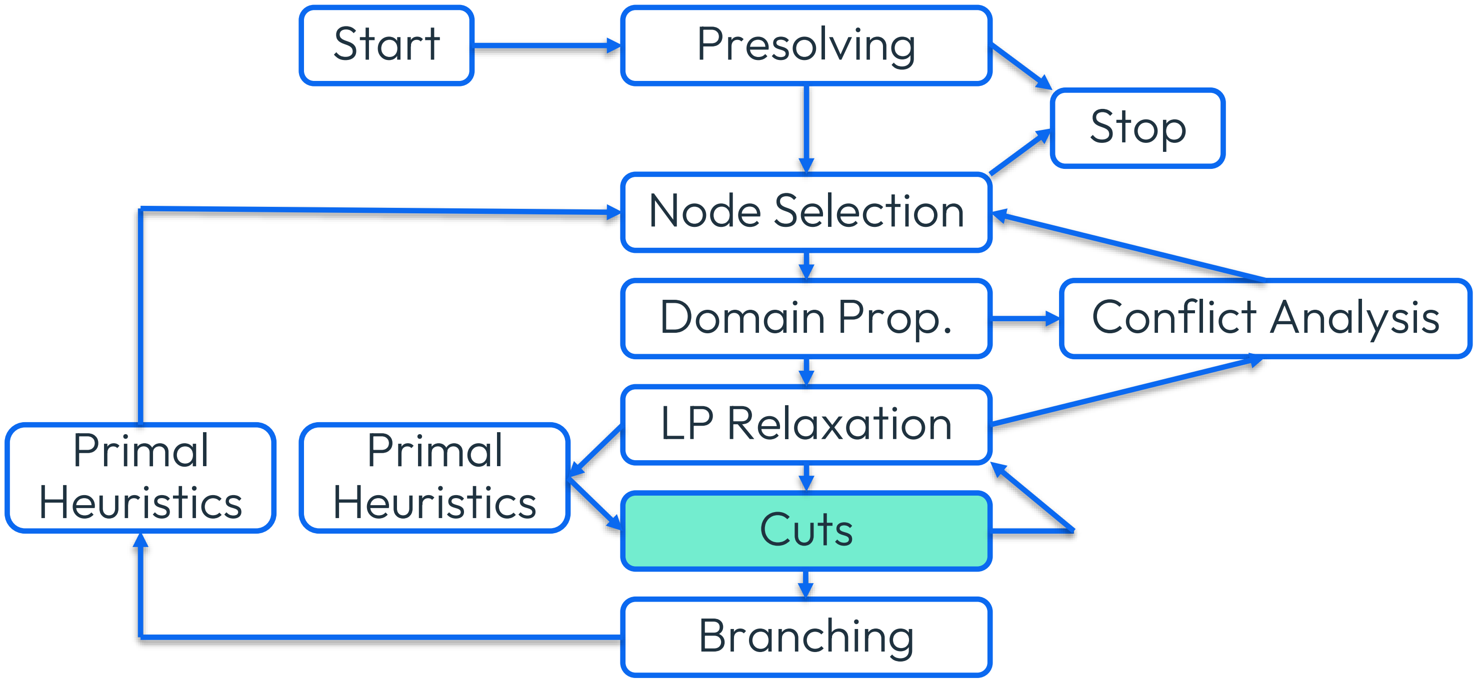
image source: DALL-E

1. Cutting plane method
2. Generic (matrix-based) cuts
3. Structure-specific cuts
4. Cut selection

Motivation

- Original MIP formulation can almost always be improved
  - Smaller difference between space of feasible continuous and feasible integer solutions
- Two techniques: next lecture
  - Presolving: Logic reductions of the model before the main search starts
  - Cutting planes: Generating additional constraints that tighten the formulation
- Three principles occur at many places in cutting and presolving:
  - Rounding: Integer multiples of integer variables take integer values
  - Lifting: Fixing a variable at a bound can make constraints infeasible or redundant
  - Disjunction: Binary variable must take one of two values

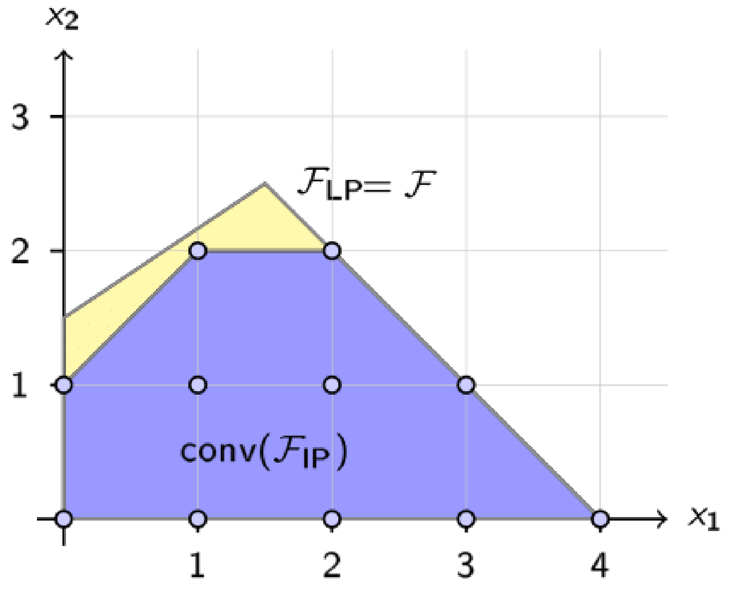
MIP Solver Flowchart



The cutting plane method

General cutting plane method: Colorful picture

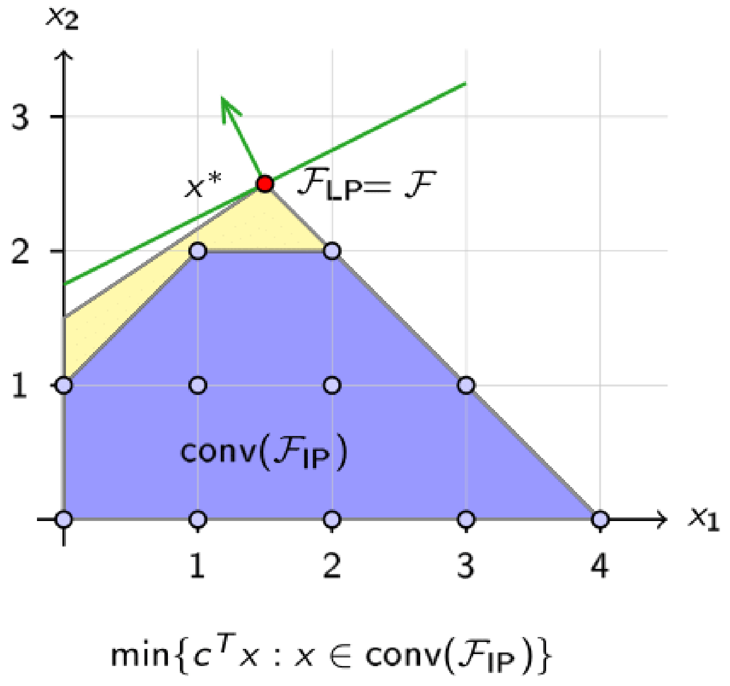
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in conv(\mathcal{F}_{IP})\}$

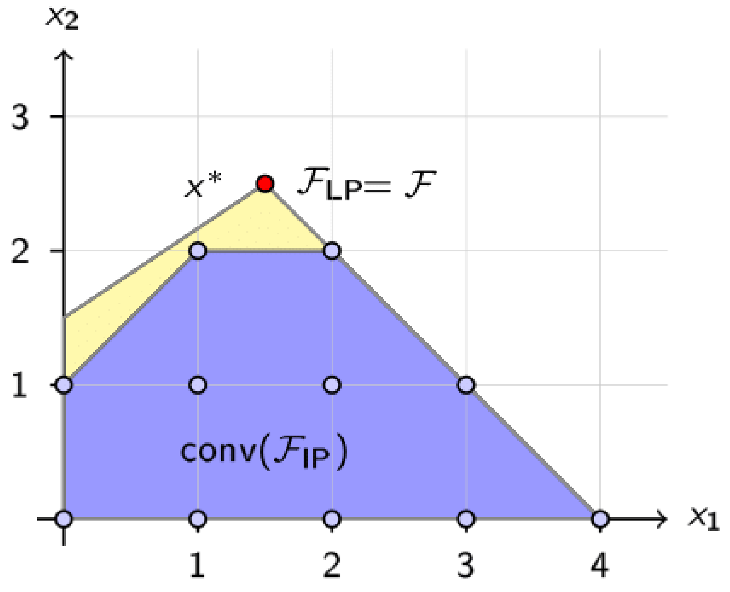
General cutting plane method: Colorful picture

1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



General cutting plane method: Colorful picture

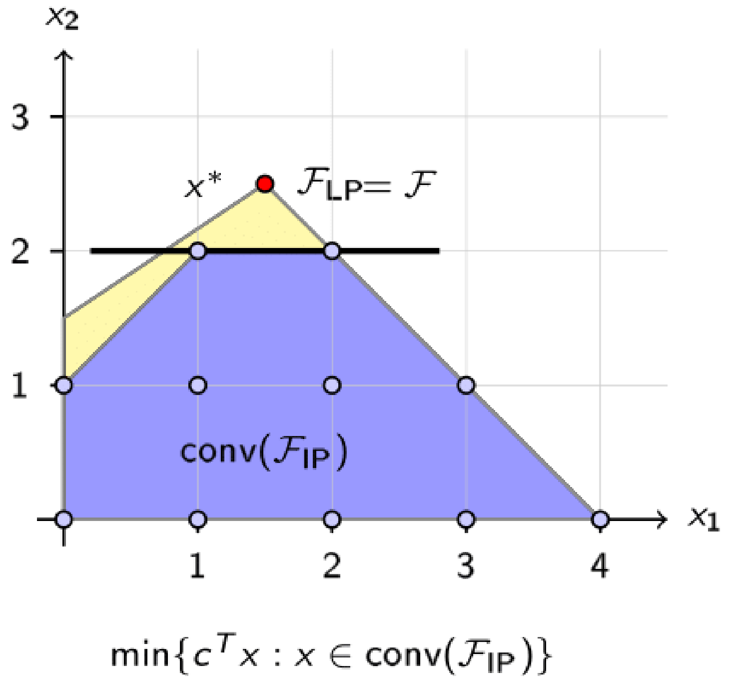
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in \text{conv}(\mathcal{F}_{IP})\}$

General cutting plane method: Colorful picture

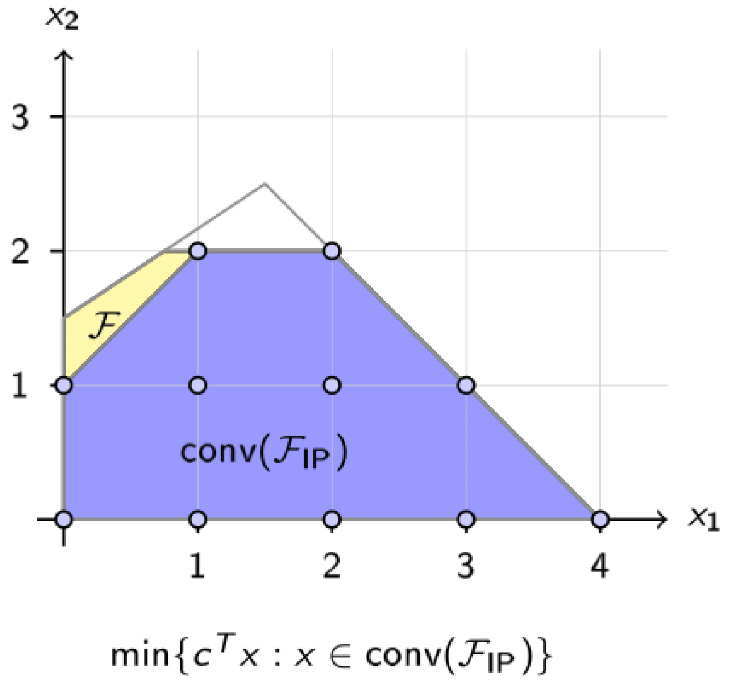
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



FICO

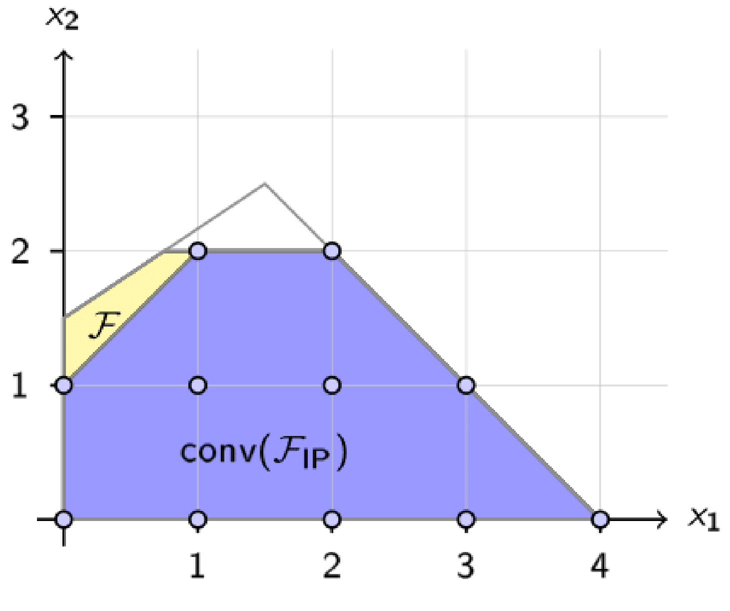
General cutting plane method: Colorful picture

1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



General cutting plane method: Colorful picture

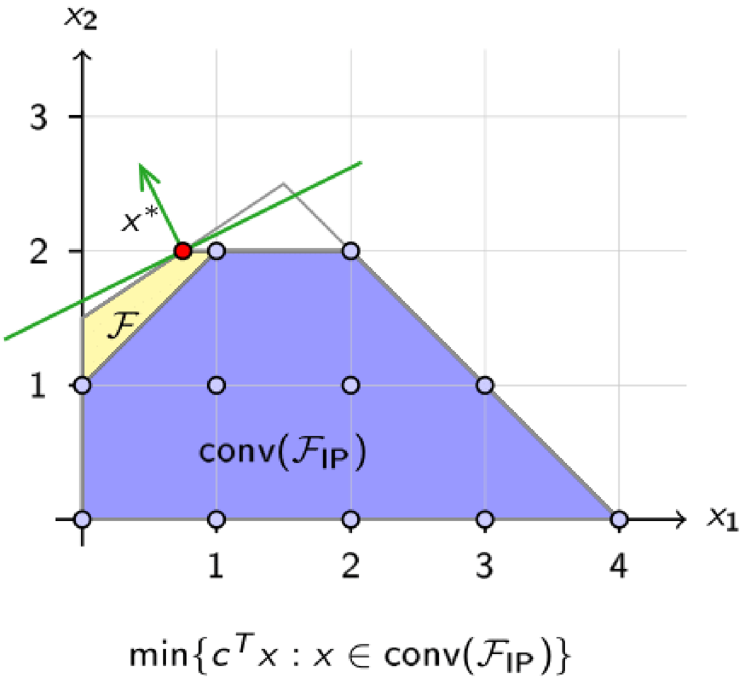
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in \text{conv}(\mathcal{F}_{\text{IP}})\}$

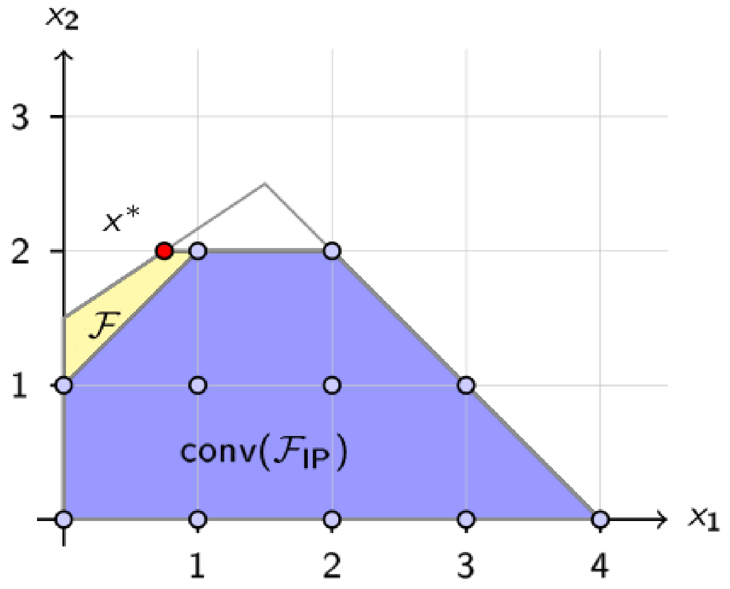
General cutting plane method: Colorful picture

1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



General cutting plane method: Colorful picture

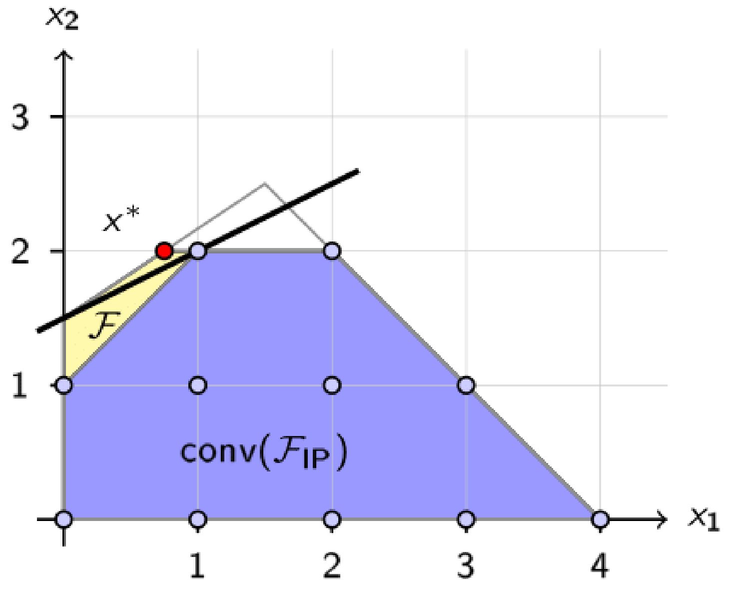
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in \text{conv}(\mathcal{F}_{IP})\}$

General cutting plane method: Colorful picture

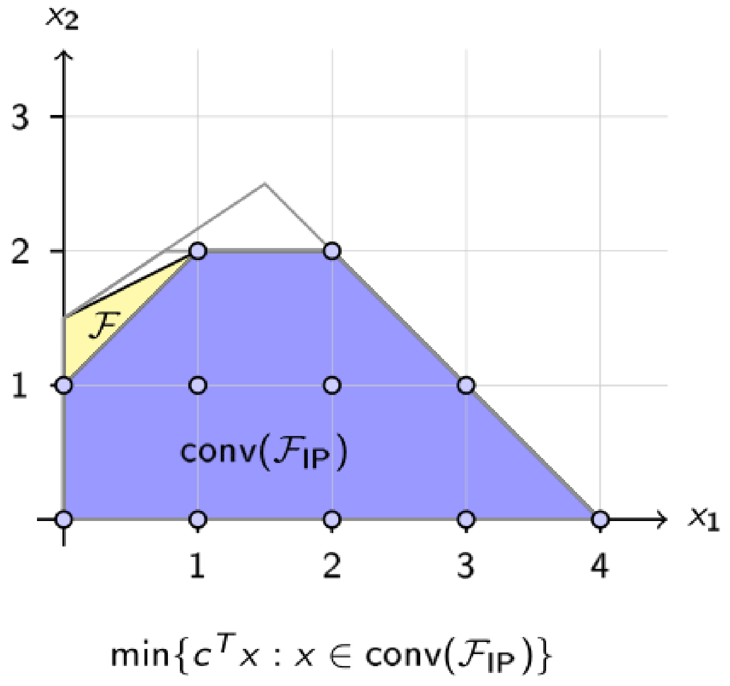
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in \text{conv}(\mathcal{F}_{IP})\}$

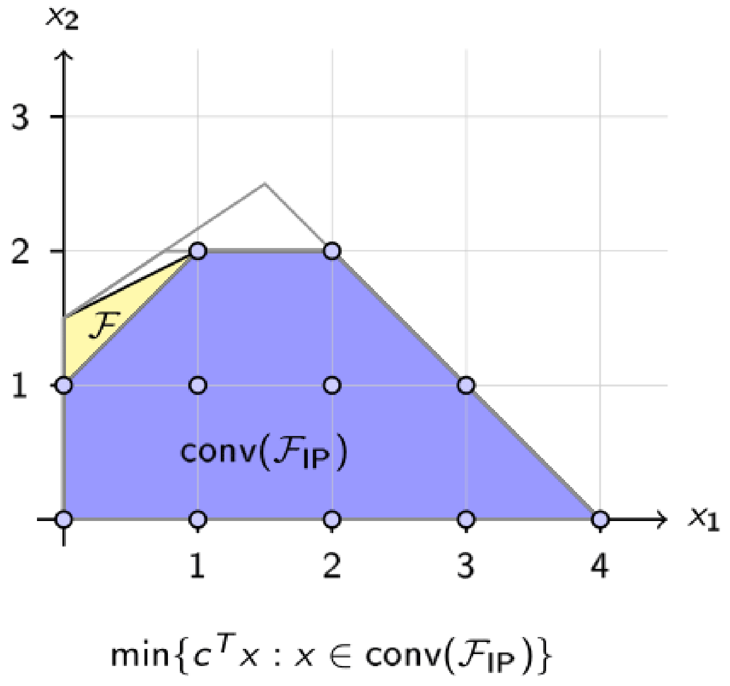
General cutting plane method: Colorful picture

1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



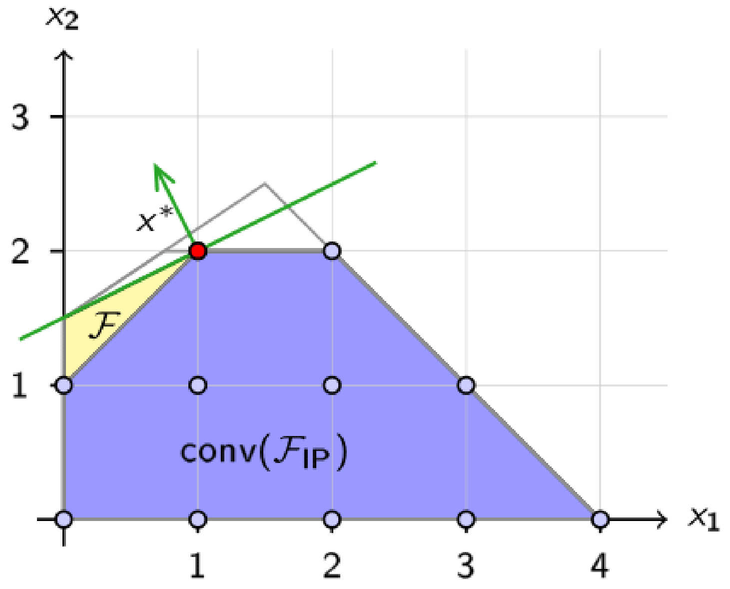
General cutting plane method: Colorful picture

1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



General cutting plane method: Colorful picture

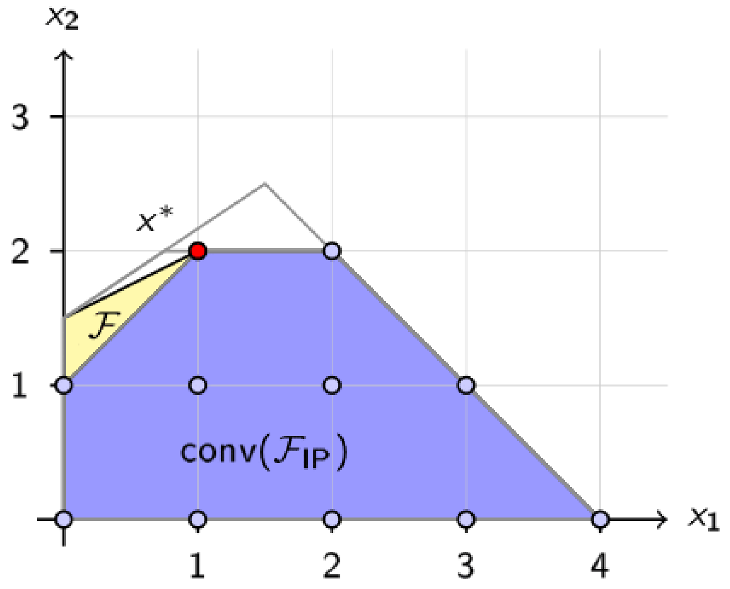
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in \mathrm{conv}(\mathcal{F}_{IP})\}$

General cutting plane method: Colorful picture

1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



$\min\{c^T x : x \in \text{conv}(\mathcal{F}_{IP})\}$

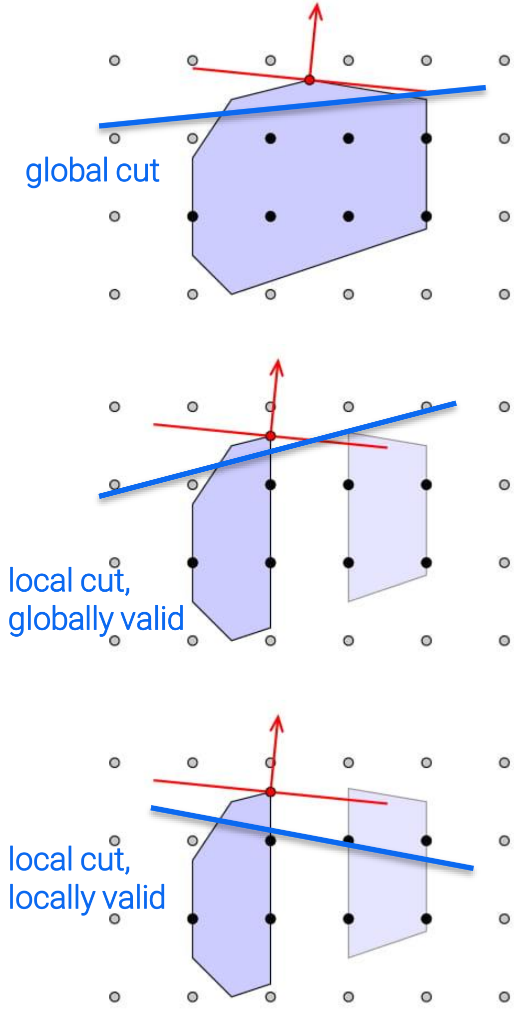
Classes of cuts

- General,“matrix-based“ cuts:
  - Gomory cuts
  - complemented MIR cuts
  - Gomory mixed integer cuts
  - strong Chvátal-Gomory cuts
  - {0, ½}-cuts
  - implied bound cuts .
  - Split cuts
  - Lift-and-project cuts
  - Mod-k cuts
  - …

- Combinatorial, „problem-specific cuts“:
  - 0-1 knapsack problem
  - stable set problem
  - 0-1 single node flow problem
  - multi-commodity-flow problem
  - …

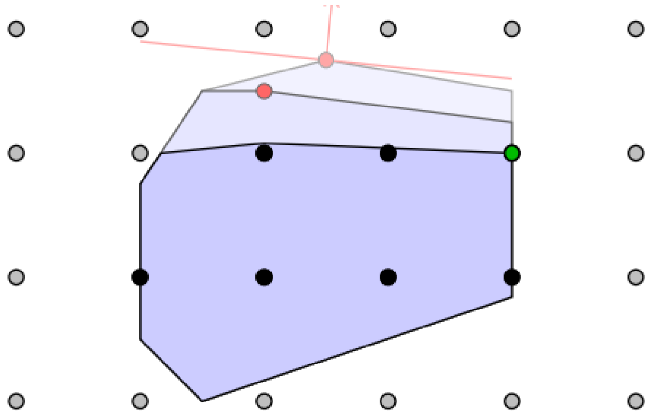
Local Cuts

- Global cuts
  - Generated at the root node
  - Hence globally valid by construction
- Local cuts
  - Generated at internal nodes
  - Either globally valid
    - When only using global information (e.g. bounds)
    - Can be re-used in other parts of the tree
  - Or locally valid
    - When using local bounds
    - Potentially stronger



Global and local cuts

- Cutting plane generation works in rounds:
  - Solve LP, remove cuts, generate cuts, filter cuts, select cuts, add cuts, repeat
- Heavily at the root node
  - Often around 20 rounds of cuts, sometimes more than 100
- Less heavy in the tree
  - Not at every node
  - Much less rounds and fewer cuts per round
  - Should we generate local cuts in the tree?
    - Locally valid or globally valid?



Matrix-based cuts

Gomory & friends

Gomory cuts: the first IP solver

OUTLINE OF AN ALGORITHM FOR INTEGER SOLUTIONS TO LINEAR PROGRAMS

BY RALPH E. GOMORY[1]

Communicated by A. W. Tucker, May 3, 1958

The problem of obtaining the best integer solution to a linear program comes up in several contexts. The connection with combinatorial problems is given by Dantzig in [1], the connection with problems involving economies of scale is given by Markowitz and Manne [3] in a paper which also contains an interesting example of the effect of discrete variables on a scheduling problem. Also Dreyfus [4] has discussed the role played by the requirement of discreteness of variables in limiting the range of problems amenable to linear programming techniques.

It is the purpose of this note to outline a finite algorithm for obtaining integer solutions to linear programs. The algorithm has been programmed successfully on an E101 computer and used to run off the integer solution to small (seven or less variables) linear programs completely automatically.

image source: ralphgomory.com

Gomory cuts (1958)

- Given an arbitrary IP, with an optimal basic solution of its LP relaxation
  - Finds for each fractional variable in the LP solution a hyperplane that separates the LP solution from the set of all feasible solutions of the IP
  - Add one (or all) to the LP relaxation, rinse, repeat
  - Assumes standard form $\max\{c^T x| \ Ax = b; x \geq 0; x \in \mathbb{Z}^n\}$
- Use basic representation of the solution
  - $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i$
    - Basic LP solution: $x_i = \bar{b}_i, x_j = 0$
    - Choose a fractional basic variable: $x_i = \bar{b}_i \notin \mathbb{Z}$

Gomory cuts, proof

- $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i \notin \mathbb{Z}$

Gomory cuts, proof

- $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i \notin \mathbb{Z}$
- Add some zeros
  - $x_i + \sum(\bar{a}_{ij} + \lfloor \bar{a}_{ij} \rfloor - \lfloor \bar{a}_{ij} \rfloor) x_j = \bar{b}_i + \lfloor \bar{b}_i \rfloor - \lfloor \bar{b}_i \rfloor$

Gomory cuts, proof

- $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i \notin \mathbb{Z}$
- Add some zeros
  - $x_i + \sum (\bar{a}_{ij} + \lfloor \bar{a}_{ij} \rfloor - \lfloor \bar{a}_{ij} \rfloor) x_j = \bar{b}_i + \lfloor \bar{b}_i \rfloor - \lfloor \bar{b}_i \rfloor$
- Sort by integral and fractional parts
  - $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$

Gomory cuts, proof

- $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i \notin \mathbb{Z}$
- Add some zeros
  - $x_i + \sum(\bar{a}_{ij} + \lfloor \bar{a}_{ij} \rfloor - \lfloor \bar{a}_{ij} \rfloor) x_j = \bar{b}_i + \lfloor \bar{b}_i \rfloor - \lfloor \bar{b}_i \rfloor$
- Sort by integral and fractional parts
  - $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum(\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$
- The left hand side must be integer for all integer solutions (and so must the right-hand side)

Gomory cuts, proof

- $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i \notin \mathbb{Z}$
- Add some zeros
  - $x_i + \sum (\bar{a}_{ij} + \lfloor \bar{a}_{ij} \rfloor - \lfloor \bar{a}_{ij} \rfloor) x_j = \bar{b}_i + \lfloor \bar{b}_i \rfloor - \lfloor \bar{b}_i \rfloor$
- Sort by integral and fractional parts
  - $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$
- The left hand side must be integer for all integer solutions (and so must the right-hand side)
- The right hand side is less than one
  - $\bar{b}_i - \lfloor \bar{b}_i \rfloor$ is less than one
  - $\sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$ is a sum of non-negative values

Gomory cuts, proof

- $x_i + \sum \bar{a}_{ij} x_j = \bar{b}_i \notin \mathbb{Z}$
- Add some zeros
  - $x_i + \sum(\bar{a}_{ij} + \lfloor \bar{a}_{ij} \rfloor - \lfloor \bar{a}_{ij} \rfloor) x_j = \bar{b}_i + \lfloor \bar{b}_i \rfloor - \lfloor \bar{b}_i \rfloor$
- Sort by integral and fractional parts
  - $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum(\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$
- The left hand side must be integer for all integer solutions (and so must the right-hand side)
- The right hand side is less than one
  - $\bar{b}_i - \lfloor \bar{b}_i \rfloor$ is less than one
  - $\sum(\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$ is a sum of non-negative values
- Hence, the right hand side must be less equal zero for all integer solutions

Gomory cuts, proof part II

- $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$
  - Right hand side must be less equal zero for all integer solutions

Gomory cuts, proof part II

- $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$
  - Right hand side must be less equal zero for all integer solutions
- Hence, $-\sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq \lfloor \bar{b}_i \rfloor - \bar{b}_i$ $(\Leftrightarrow \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq 0)$
  is a valid inequality for the given IP

Gomory cuts, proof part II

- $x_i + \sum\lfloor\bar{a}_{ij}\rfloor x_j - \lfloor\bar{b}_i\rfloor = \bar{b}_i - \lfloor\bar{b}_i\rfloor - \sum(\bar{a}_{ij} - \lfloor\bar{a}_{ij}\rfloor)x_j$
  - Right hand side must be less equal zero for all integer solutions
- Hence, $-\sum(\bar{a}_{ij} - \lfloor\bar{a}_{ij}\rfloor)x_j \leq \lfloor\bar{b}_i\rfloor - \bar{b}_i$ $\quad (\Leftrightarrow \bar{b}_i - \lfloor\bar{b}_i\rfloor - \sum(\bar{a}_{ij} - \lfloor\bar{a}_{ij}\rfloor)x_j \leq 0)$
  is a valid inequality for the given IP
  - And it is violated by the basic LP solution, since for the initial LP solution, the left-hand side is zero, but the right-hand side is negative

Gomory cuts, proof part II

- $x_i + \sum \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j$
  - Right hand side must be less equal zero for all integer solutions
- Hence, $-\sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq \lfloor \bar{b}_i \rfloor - \bar{b}_i$ $(\Leftrightarrow \bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq 0)$
  is a valid inequality for the given IP
  - And it is violated by the basic LP solution, since for the initial LP solution, the left-hand side is zero, but the right-hand side is negative
- This is the Gomory cut!
- Add a slack variable, add to the equation system, iterate
- Similar idea works for mixed-integer programming (Gomory 1960)

Chvátal-Gomory (Chvátal 1973)

- Works on original matrix. Only for pure integer constraints.
- Let A_j be the j-th column of A and $\lambda \in \mathbb{R}^m_{\geq 0}$
- Aggregate: $\sum \lambda A_j x_j \leq \lambda b$
- Rounding, step 1: $\sum \lfloor \lambda A_j \rfloor x_j \leq \lambda b$
  - Valid, since $\sum \lfloor \lambda A_j \rfloor x_j \leq \sum \lambda A_j x_j$ and $x_j \geq 0$
  - Relaxation
- Rounding, step 2: $\sum \lfloor \lambda A_j \rfloor x_j \leq \lfloor \lambda b \rfloor$
  - Valid, since $x \in \mathbb{Z}^n$
  - Strengthening

Chvátal-Gomory Example

- $5x + 5y \leq 9$ (I)
  $7x + 3y \leq 9$ (II)

- Aggregate: $\sum \lambda A_j x_j \leq \lambda b$

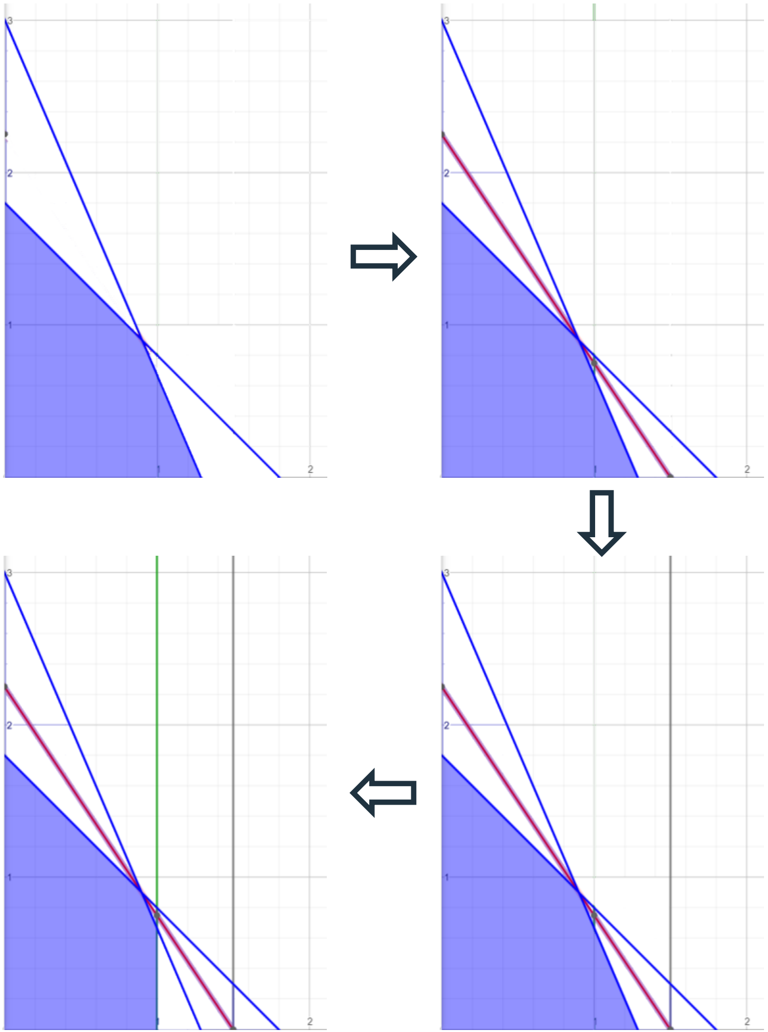
  $x + \frac{2}{3}y \leq \frac{3}{2}$ $\quad \frac{1}{12}$(I)+ $\frac{1}{12}$(II)

- Rounding, step 1: $\sum \lfloor \lambda A_j \rfloor x_j \leq \lambda b$

  $x \leq \frac{3}{2}$

- Rounding, step 2: $\sum \lfloor \lambda A_j \rfloor x_j \leq \lfloor \lambda b \rfloor$

  $x \leq 1$

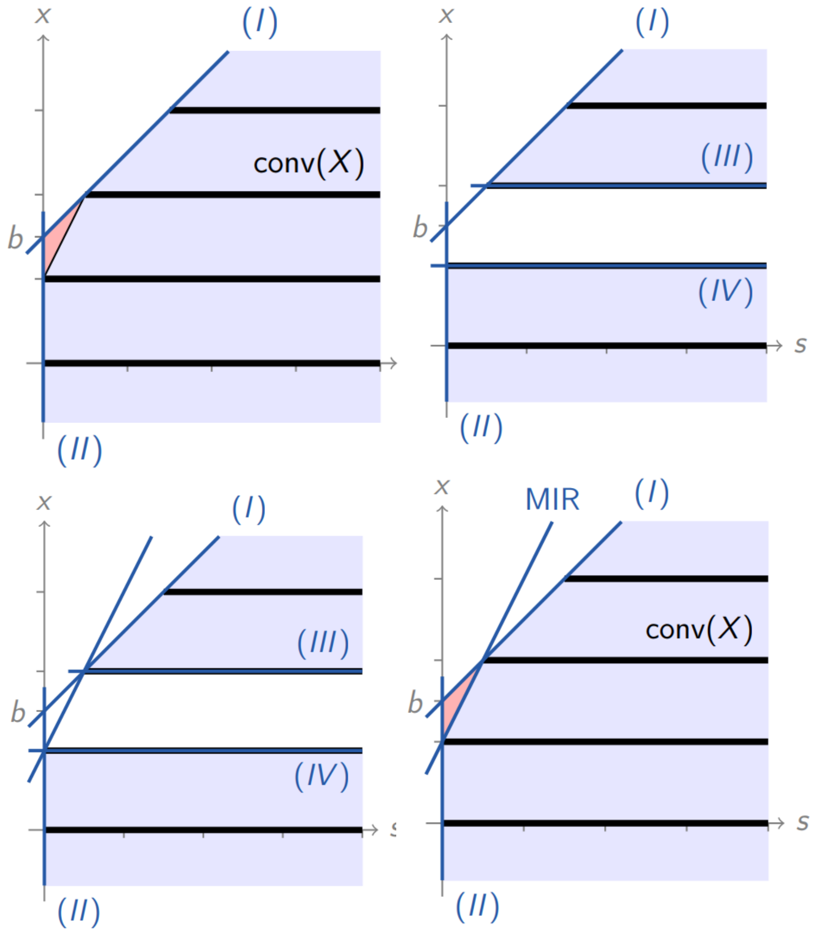


{0, ½} and mod-k (Caprara&Fischetti 1996, Caprara et al 2000)

- How to choose λ for Chvátal-Gomory cuts?
  - Many heuristics exist…
  - λ can be replaced by λ- $\lfloor\lambda\rfloor \in [0,1)^m$
- Important special case: $\lambda \epsilon \{0, \frac{1}{2}\}^m$
  - For subclasses of {0,½}-cuts, there are efficient algorithms to compute strongest cut
- Many important sets of facet-defining inequalities can be expressed as {0,½}-cuts
  - Odd cycle inequalities for stable set
  - Comb inequalities for TSP
  - Blossom inequalities for b-matching
- Generalization: mod-k cuts with $\lambda \epsilon \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}\}^m$

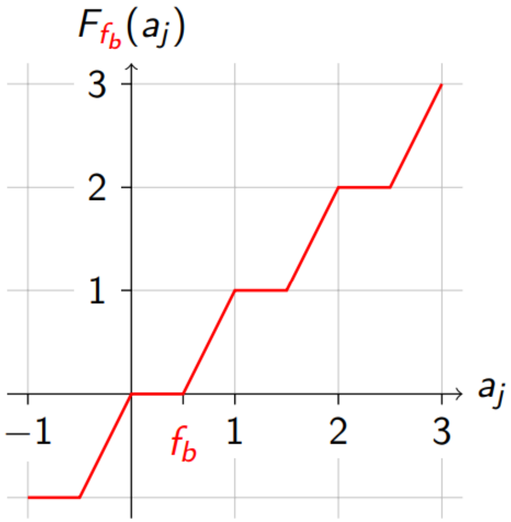
Mixed-Integer Rounding (MIR)

- Mixed-Integer set:
  - $X \;:= \{(x,s) \in \mathbb{Z} \times \mathbb{R} \colon x \le b + s$ (I), $\; s \ge 0$ (II)$\}$
  - Inequalities do not suffice to describe conv(X)
- Disjunctive Argument:
  - Here: $x \ge \lceil b \rceil$ (III) and $x \le \lfloor b \rfloor$ (IV)
  - If an inequality is valid for X_1 and for X_2, it is also valid for $X_1 \cup X_2$
- Simple MIR inequality: $x \le \lfloor b \rfloor + \frac{s}{1-(b-\lfloor b \rfloor)}$
  - This is (I) $+ (b - \lfloor b \rfloor)$(III)
  - This is (II) $+ (1 - (b - \lfloor b \rfloor))$(IV)



Complemented MIR

- Mixed knapsack set
  - $X := \{(x,s) \in \mathbb{Z}_+^n \times \mathbb{R}_+ : \sum a_j x_j \le b + s, x_j \le u_j\}$
  - General MIR inequality:
    $\sum a_j x_j \le \lfloor b \rfloor + \frac{s}{1-f_b}$ with $f_b = (b - \lfloor b \rfloor)$
- Even slightly more general MIR inequality:
  - $X := \{(x,s) \in \mathbb{Z}_+^{n_1} \times \mathbb{R}_+^{n_2} : \sum a_j x_j \le b + \sum s_k, x_j \le u_j\}$
  - $\sum a_j x_j \le \lfloor b \rfloor + \frac{\sum s_k}{1-f_b}$ with $f_b = (b - \lfloor b \rfloor)$
- c-MIR inequality:
  - Divide by a positive δ (typically integer multiple of some a_j)
  - Complement some of the integers ($x_j = u_j - \bar{x}_j$)
  - $\sum F_f(a_j) x_j \le \lfloor b \rfloor - \frac{s}{1-f_b}$
- Many classes of cuts are c-MIR cuts of aggregations of the constraint matrix



Quiz time

- Mixed-Integer Rounding Cuts
  a) Rely on a disjunctive argument
  b) Are less powerful than Mixed-Integer Gomory Cuts
  c) Require the solution of an Auxiliary LP
- $\{0, {}^1\!/_2\}$-cuts work
  a) On a graph structure
  b) On the original constraint matrix
  c) On the Simplex tableau
- In MIP solvers, cut generation is typically applied
  a) Only at the root node
  b) Aggressively at the root and moderately at some tree nodes
  c) The same way at all nodes

Quiz time

- Mixed-Integer Rounding Cuts
  a) Rely on a disjunctive argument
  b) Are less powerful than Mixed-Integer Gomory Cuts
  c) Require the solution of an Auxiliary LP
- $\{0, {}^1/_2\}$-cuts work
  a) On a graph structure
  b) On the original constraint matrix
  c) On the Simplex tableau
- In MIP solvers, cut generation is typically applied
  a) Only at the root node
  b) Aggressively at the root and moderately at some tree nodes
  c) The same way at all nodes

Quiz time

- Mixed-Integer Rounding Cuts
  a) Rely on a disjunctive argument
  b) Are less powerful than Mixed-Integer Gomory Cuts
  c) Require the solution of an Auxiliary LP
- $\{0, {}^{1}/_{2}\}$-cuts work
  a) On a graph structure
  b) On the original constraint matrix
  c) On the Simplex tableau
- In MIP solvers, cut generation is typically applied
  a) Only at the root node
  b) Aggressively at the root and moderately at some tree nodes
  c) The same way at all nodes

Quiz time

- Mixed-Integer Rounding Cuts
  a) Rely on a disjunctive argument
  b) Are less powerful than Mixed-Integer Gomory Cuts
  c) Require the solution of an Auxiliary LP
- $\{0, {}^{1}/_{2}\}$-cuts work
  a) On a graph structure
  b) On the original constraint matrix
  c) On the Simplex tableau
- In MIP solvers, cut generation is typically applied
  a) Only at the root node
  b) Aggressively at the root and moderately at some tree nodes
  c) The same way at all nodes

Structure-specific cuts

Let the combinatorics out!

Knapsack Cover cuts (Balas & Zemel 1978)

- Feasible set of knapsack problem: $X^K := \{x \in \{0,1\}^n : \sum a_j x \leq b\}$ with $(b \in \mathbb{Z}_+, a_j \in \mathbb{Z}_+)$
- Cover: subset C of the variables s.t.
  - $\sum_{j \in C} a_j > b$
- Minimal Cover: subset C of the variables s.t.
  - $\sum_{j \in C} a_j > b$
  - $\sum_{j \in C \setminus \{i\}} a_j \leq b$ for all $i \in C$
- Minimal Cover Inequality
  - $\sum_{j \in C} x_j \leq |C| - 1$
- Example: $5x_1 + 6x_2 + 2x_3 + 2x_4 \leq 8$
  - Minimal cover: $C = \{2,3,4\}$
  - Minimal cover inequality: $x_2 + x_3 + x_4 \leq 2$

FICO

Lifting cover inequalities

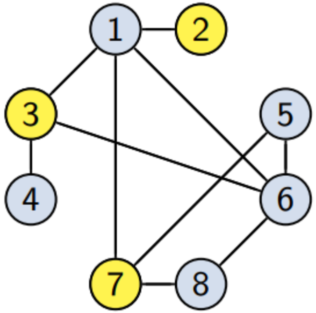
- Can we incorporate variable x_1 in our cover cut $x_2 + x_3 + x_4 \leq 2$?
  - Is there an α_i s.t. $\alpha_i x_i + \sum_{j \in C} x_j \leq |C| - 1,\ i \notin$ C is a valid inequality for X^K ?
- Disjunctive argument:
- $\alpha_i x_i + \sum_{j \in C} x_j \leq |C| - 1$ is valid for $X^K \cap \{x \in \{0,1\}^n : x_i = 0\}$ for all α_i
- $\alpha_i x_i + \sum_{j \in C} x_j \leq |C| - 1$ is valid for $X^K \cap \{x \in \{0,1\}^n : x_i = 1\}$

  $\Leftrightarrow \alpha_i \cdot 1 + \max\left\{\sum_{j \in C} x_j : \sum a_j x \leq b, x \in \{0,1\}^n, x_i = 1\right\} \leq |C| - 1$

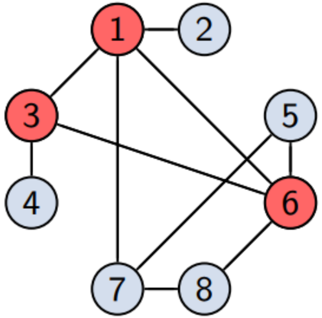
  $\Leftrightarrow \alpha_i \leq |C| - 1 - \max\left\{\sum_{j \in C} x_j : \sum a_j x \leq b, x \in \{0,1\}^n, x_i = 1\right\}$
- $\alpha_1 x_1 + x_2 + x_3 + x_4 \leq 2$ is valid for $\{x \in \{0,1\}^4 : 5x_1 + 6x_2 + 2x_3 + 2x_4 \leq 8\}$
  - $\Leftrightarrow \alpha_1 \leq 2 - \max\{x_2 + x_3 + x_4 : 6x_2 + 2x_3 + 2x_4 \leq 3, x \in \{0,1\}^4\} \Leftrightarrow \alpha_1 \leq 1$
  - $\Rightarrow x_1 + x_2 + x_3 + x_4 \leq 2$ is a valid inequality!

The stable set problem (Chvátal 1975)

- Given a graph $G = (V, E)$. A stable set is a set of non-adjacent vertices
  - Stable Set: $S \subseteq V$, for all $u, v \in S : (u, v) \notin E$
- Stable set polytope for graph $G = (V, E)$:
  - $conv(\{x \in \{0,1\}^{|V|} : x_u + x_v \leq 1 \; for \; all \; (u, v) \in E\})$
- Given a graph $G = (V, E)$. A clique is a set of pairwise adjacent vertices
  - Clique: $S \subseteq V$, for all $u, v \in S : (u, v) \in E$
- Clique inequalities: $\sum_{j \in C} x_j \leq 1$
  - Valid for stable set polytope



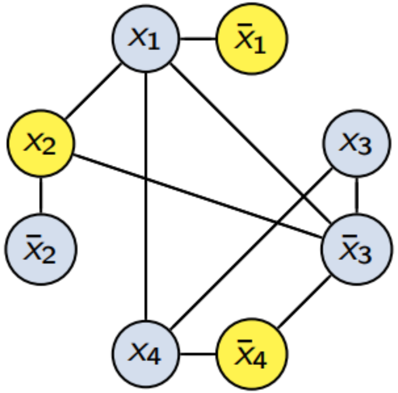
Stable set



Clique

Cutting from the clique table/graph

- Clique Graph: A graph $G = (V, E)$.
  - A node for every binary variable x_j and for its complement $\bar{x}_j := 1 - x_j$
  - Add an edge $(x_i, x_j) \in E$ whenever we find that for all feasible MIP solutions, x_i and x_j cannot be one at the same time.
    - Can come from assignment constraints $\sum x_i = 1$, but also from 2-elementary knapsack covers, or from constraints such as $x_1 + x_2 + x_3 \geq 2$, from probing,…
- Feasible MIP solution corresponds to stable set in clique graph
- Stable set polytope of the conflict graph is a relaxation of the MIP's feasible region
- Separation algorithm: Find maximal violated cliques in clique graph
  - Heuristic / greedy DFS tree search



Cut Selection

Luxury problem: Which cuts should we use?

- How many cuts should be generated for a relaxation solution?
  - One?
    - Will provide a new relaxation solution
    - Expensive to re-solve relaxation for each cut
  - As many as possible?
    - Relaxation solution only needs to be cut off once
    - Cuts increase the size of the model
    - Cutting plane separators might be expensive
- Balancing is important:
  - Multiple rounds, limited number of cuts per round, replace old with new ones
  - Carefully choose which cuts complement each other nicely

Cut selection: What does a good cut look like?

- Numerically stable:
  - Coefficient range not too large, neither the absolute values
  - Hard criterion, throw cuts away that fail this
- Efficient:
  - Distance of hyperplane to the LP solution, cut as deeps as possible into the polyhedron
  - Soft criterion, minimum efficacy should be met
- Orthogonal w.r.t. other cuts
  - Ideally, pairwise almost orthogonal, each cut „cuts off a different part of the polyhedron“
- Almost parallel to the objective:
  - Exactly parallel is bad (degeneracy!), throw cut away (and only use as dual bound)
  - Almost parallel should trigger progress in dual bound

FICO

Cut selection: How does a good cut look like? (ctd)

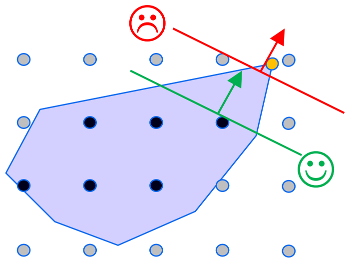
- Sparse: Only a few (integer) variables
- Dcd: Cutting towards primal solution (use directed distance between LP opt and incumbent)

Selection process:

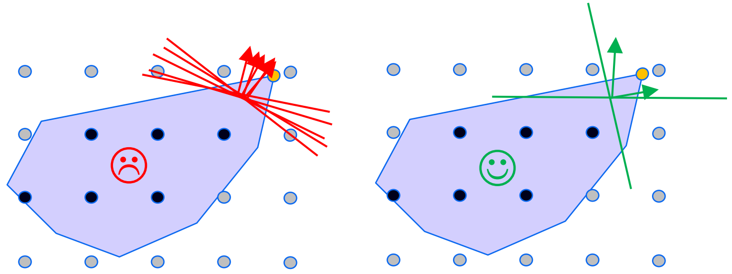
- Aggregate different measures and compute a single score
- Greedily select cut with highest score, remove similar cuts, iterate until no cut left or maximum number of cuts / cut elements hit

Cut selection: What is a good cut?

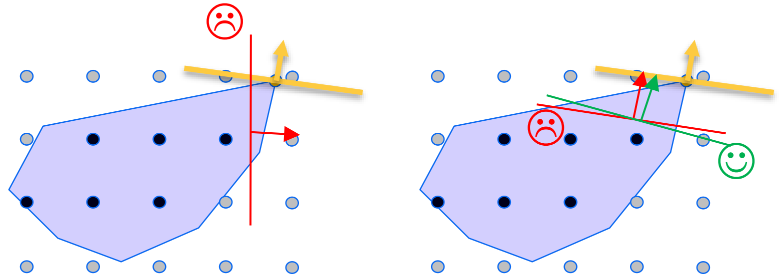
- Efficacy (Deep is good)



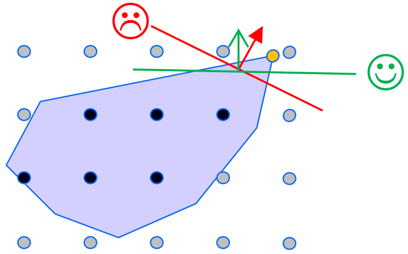
- Orthogonality (Different is good)



- Obj-parallelism (Similar is good)



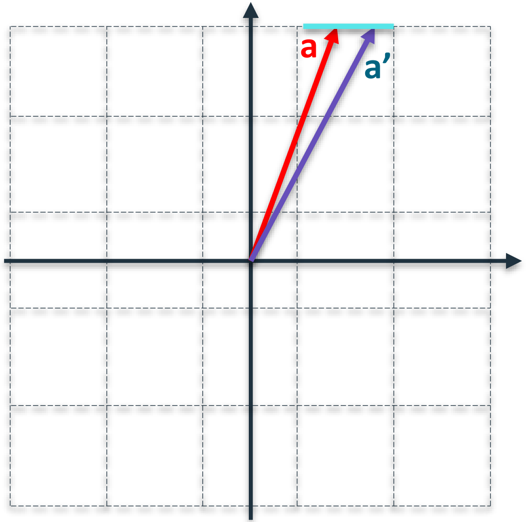
- Sparsity (Few dimensions are good)



Unit cube hashing to estimate orthogonality

- Calculating orthogonality for all cut pairs can be too expensive
  - Complexity is $O(n^2m)$ in the number of cuts n and of non-zero cut elements m
- Unit cube cut hashing:
  1. Normalize the cut $a\,x \leq b$ to have $||a||_\infty = 1$
  2. Partition the interval in [-1,1] into 2k+1 subintervals
  3. Map each coefficient a_j in [-1,1] to an integer d_j in {-k, …, 0, …, k}
  4. Hash the discretized cut vector **d**
- Use unit ball hashing to skip many orthogonality calculations
  - Two cuts with different hash code are considered different enough
    - Likely to be different enough at least in one coefficient
  - Two cuts with same hash code might be similar
    - Compute the actual orthogonality factor between the cuts

K = 2   a = (2.25, 6.25)   a' = (2.6, 5)



Thank You!

Timo Berthold
TU Berlin, FICO, MODAL

image source: DALL-E

Knapsack cover cuts: Little Exercise

- Consider the knapsack problem $\max\{\mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3 \mid 3x_1 + 4x_2 + 7x_3 + 9x_4 \leq 11, x \in \{0,1\}^4\}$
  - What is the optimal solution vector?
  - Find a knapsack cover cut that cuts this solution off
  - Can we lift another variable into that cut?

Knapsack cover cuts: Little Exercise

- Consider the knapsack problem $\max\{\mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3 \mid 3x_1 + 4x_2 + 7x_3 + 9x_4 \leq 11, x \in \{0,1\}^4\}$
  - What is the optimal solution vector?
  - Find a knapsack cover cut that cuts this solution off
  - Can we lift another variable into that cut?

  - The optimal LP solution is $\mathrm{x}^* = (1, 1, \frac{4}{7}, 0)$ with a solution value of $2\,\frac{4}{7}$.

Knapsack cover cuts: Little Exercise

- Consider the knapsack problem $\max\{x_1 + x_2 + x_3 \mid 3x_1 + 4x_2 + 7x_3 + 9x_4 \leq 11, x \in \{0,1\}^4\}$
  - What is the optimal solution vector?
  - Find a knapsack cover cut that cuts this solution off
  - Can we lift another variable into that cut?

- The optimal LP solution is $x^* = (1, 1, \frac{4}{7}, 0)$ with a solution value of $2\frac{4}{7}$.
- $\{1,2,3\}$ is a cover, since $3 + 4 + 7 = 14 > 11$, but $3 + 4$ and $3 + 7$ and $4 + 7$ are all ≤ 11.
  - $x_1 + x_2 + x_3 \leq 2$ is a cover cut
  - Since $x_1^* + x_2^* + x_3^* = 2\frac{4}{7} > 2$, it cuts off the LP optimum

FICO

Knapsack cover cuts: Little Exercise

- Consider the knapsack problem $\max\{x_1 + x_2 + x_3 \mid 3x_1 + 4x_2 + 7x_3 + 9x_4 \leq 11, x \in \{0,1\}^4\}$
  - What is the optimal solution vector?
  - Find a knapsack cover cut that cuts this solution off
  - Can we lift another variable into that cut?

  - The optimal LP solution is $x^* = (1, 1, \frac{4}{7}, 0)$ with a solution value of $2\frac{4}{7}$.
  - $\{1,2,3\}$ is a cover, since $3 + 4 + 7 = 14 > 11$, but $3 + 4$ and $3 + 7$ and $4 + 7$ are all ≤ 11.
    - $x_1 + x_2 + x_3 \leq 2$ is a cover cut
    - Since $x_1^* + x_2^* + x_3^* = 2\frac{4}{7} > 2$, it cuts off the LP optimum
  - Only x_4 remains. Lift it with any $\alpha_4 \leq 2 - \max\{x_1 + x_2 + x_3 | 3x_1 + 4x_2 + 7x_3 \leq 2\} = 2$
  - $x_1 + x_2 + x_3 + 2x_4 \leq 2$ is a valid inequality