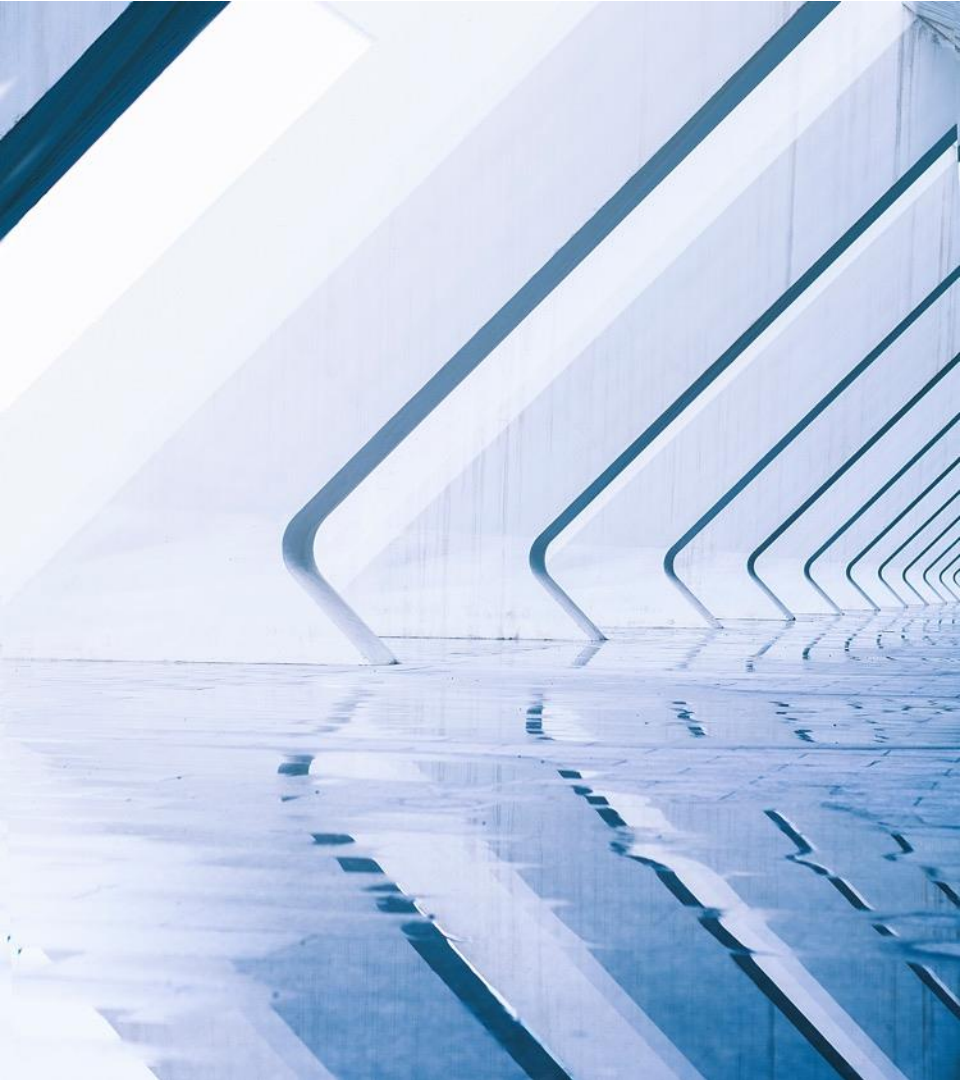


FICO

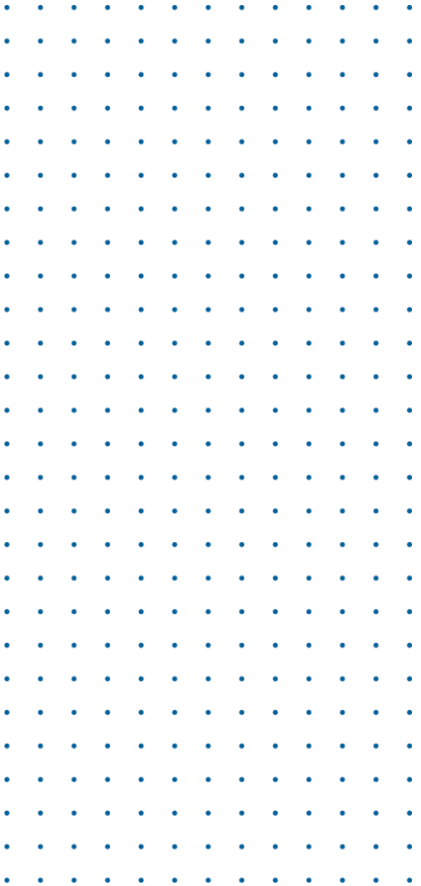
Theory and Basics

A glimpse of the big three

Timo Berthold

Agenda for this lecture

- Simplex algorithm
- Ellipsoid method, Barrier
- Gomory cuts
- Branch&Bound



Simplex Algorithm

Click to add text

Simplex idea

- Start at a random vertex
- Among neighboring vertices, choose one which improves the objective, if none: optimal
  - Special case: A ray along which the objective improves
    - Then, the LP is unbounded
- move along edges towards an optimal vertex
  Animation: https://www.youtube.com/watch?v=54blxYi5JF8
- Why does this work?
  1. Show that objective function is optimized at the boundary of the polyhedron (convexity)
  2. Narrow down further that it is optimized in a vertex (linearity)
  3. Local optimality is global optimality (convexity)

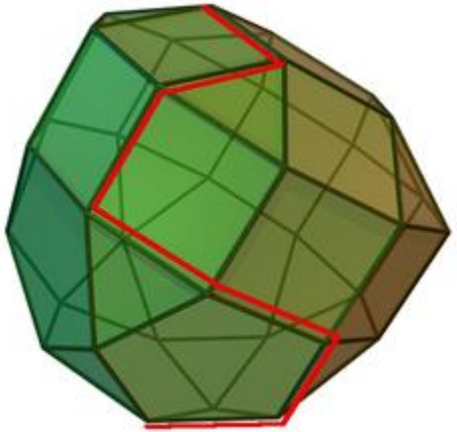


image source: WIkipedia

Basic solution

W.l.o.g., let $\text{rank}(A) = m < n$.
For every vertex $\tilde{x}$ of the LP's polyhedron

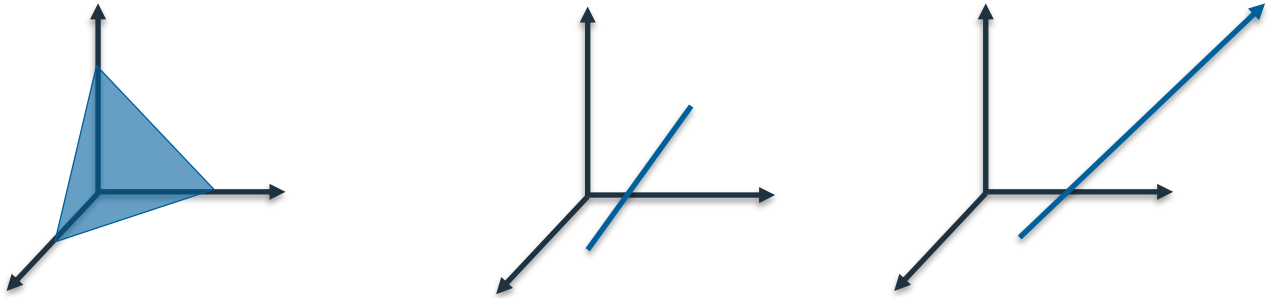
$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \in R^n_{\geq 0} \end{aligned}$$

there is a non-singular (m, m)-submatrix A_B (called basis) of A representing the vertex $\tilde{x}$ (basic solution) as follows:

$$\tilde{x}_B = A_B^{-1} b, \quad \tilde{x}_N = 0$$

Standard form

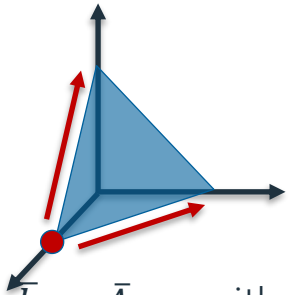
- LPs $\max\{c^T x \mid Ax = b, x \geq 0\}$ are called LPs in standard form
  - Add slack variables
- Nonempty LPs in standard form always have vertices
  - Intersection of a simplicial cone and an affine subspace



- Further, simplex algo assumes rank(A)=m<n, $\{x \mid Ax = b\}$ nonempty

Simplex idea (reloaded)

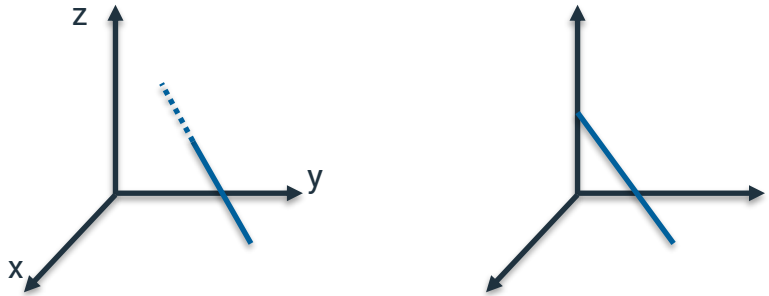
- LP optimal ⇔ there is an optimal vertex solution ⇔ there is an optimal basic solution
  - LP is a discrete optimization task, $\binom{n}{m}$ possible bases
- Idea: Find some feasible basis, exchange one of the columns in the basis („pivoting“), s.t. feasibility is maintained, objective is improved



- $x_i = \bar{b}_i - \bar{A}_i x_N$ with nonzero element $\bar{A}_{ij}$, solve for x_j. Multiply by identity matrix E with jth column: $\frac{-1}{\bar{A}_{ij}} \bar{A}_j$ and (i,j)-th element $\frac{1}{\bar{A}_{ij}}$

Which columns to choose for pivoting?

- Choose j among all columns for which $\bar{c} = c_N - c_B A_B^{-1} A_N$ is positive
  - Reduced costs, in this direction we can improve
  - If no such j: Done, optimal
- Then, among all i with $\bar{A}_{ij} > 0$ have to choose one with minimal positive $\frac{\bar{b}_i}{\bar{A}_{ij}}$ to stay feasible
  - If no such $(\bar{A}_j \leq 0)$: Done, LP is unbounded



  - Main choice: entering nonbasic column. $\frac{\bar{b}_i}{\bar{A}_{ij}}$ „bottleneck“ by how much it can be increased

One step of the Simplex algorithm (Dantzig 1947)

1. Check optimality: $\bar{c} \leq 0$?
2. Choose pivot column with $\bar{c}_j > 0$
3. Check unboundedness: $\bar{A}_j \leq 0$?
4. Pick pivot row with minimal $\frac{\bar{b}_i}{\bar{A}_{ij}}$
5. Pivoting: Exchange i,j in N and B, multiply A_B^{-1} by eta matrix
6. Update all data structures $(\bar{A}, \bar{b}, \bar{c})$

Correctness of simplex algorihm

- If all vertices non-degenerate, simplex algorithm finds solution in finite time
  - Each step new, strictly improving basis
- Issue: Cycling (resolve by lexicographic rules or by slight pertubation)

- Initial feasible solution found by auxiliary LP (phase I), which itself has a trivial initial feasible solution
- $(A\ I)\begin{pmatrix} x \\ y \end{pmatrix} = b$ has trivial solution $x = 0, y = b$, now minimize Σy
  - Phase I LP is bounded, rank(A)=m<n
  - Final step: Pivot all auxiliary columns out of the basis

Dual simplex

- Dual solution can be computed from primal and vice versa
- Hence, we can solve the dual problem instead
  - Primal simplex on dual LP
  - Meaning: Instead of staying feasible and getting more optimal, we stay „optimal“ and get more feasible
  - Approach optimum via infeasible basic solutions

Changing your LP after optimization

- Change the objective
  - Basis stays feasible, simplex can get warm-started
- Add a column
  - Basis stays feasible (add to nonbasis), warm-start
- Add a row
  - Basis stays dual feasible (add slack to basis), warm-start dual simplex
- Change right hand side / variable bound
  - Stays dual feasible, warm-start dual simplex

Ellipsoid Method & Barrier

Click to add text

Khachiyan‘s algorithm – the first polynomial time LP solver

Доклады Академии наук СССР
1979. Том 244, № 5

УДК 519.95

МАТЕМАТИКА

Л. Г. ХАЧИЯН

ПОЛИНОМИАЛЬНЫЙ АЛГОРИТМ В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ

(Представлено академиком А. А. Дородницыным 4 X 1978)

Рассмотрим систему из $m \geqslant 2$ линейных неравенств относительно $n \geqslant 2$ вещественных переменных $x_1, \ldots, x_j, \ldots, x_n$

$$a_{i1}x_1 + \ldots + a_{in}x_n \leqslant b_i, \quad i = 1, 2, \ldots, m, \tag{1}$$

с целыми коэффициентами a_{ij}, b_i. Пусть

$$L = \left[\sum_{i,j=1}^{m,n} \log_2(|a_{ij}|+1) + \sum_{i=1}^{m} \log_2(|b_i|+1) + \log_2 nm \right] + 1 \tag{2}$$

есть длина входа системы, т. е. число символов 0 и 1, необходимых для записи (1) в двоичной системе счисления.

Mathematical Programming Study 14 (1981) 61–68.
North-Holland Publishing Company

KHACHIYAN'S ALGORITHM FOR LINEAR PROGRAMMING*

Peter GÁCS and Laszlo LOVÁSZ

Computer Science Department, Stanford University, Stanford, CA 94305, U.S.A.

Received 10 October 1979

L.G. Khachiyan's algorithm to check the solvability of a system of linear inequalities with integral coefficients is described. The running time of the algorithm is polynomial in the number of digits of the coefficients. It can be applied to solve linear programs in polynomial time.

Key Words: Linear Programming, Inequalities, Complexity, Polynomial Algorithms.

0. Introduction

L.G. Khachiyan [1, cf. also 2, 3] published a polynomial-bounded algorithm to solve linear programming. These are some notes on this paper. We have ignored his considerations which concern the precision of real computations in order to make the underlying idea clearer; on the other hand, proofs which are missing from his paper are given in Section 2. Let

$$a_i x < b_i \quad (i = 1, \ldots, m, a_i \in Z^n, b_i \in Z) \tag{1}$$

be a system of *strict* linear inequalities with integral coefficients. We present an algorithm which decides whether or not (1) is solvable, and yields a solution if it is. Define

$$L = \sum_{i,j} \log(|a_{ij}|+1) + \sum_i \log(|b_i|+1) + \log nm + 1.$$

L is a lower bound on the space needed to state the problem.

Ellipsoid method

- Idea: From coefficients in A and b, we can determine largest possible solution value for x and minimum size of polyhedron
- Find large ball, which must contain a feasible solution, if one exist
- Check whether center point is feasible
- Cut ball/ellipsoid in (less than) half, determine smallest ellipsoid that contains half ellipsoid
- Repeat until ellipsoid is so small that polytope must be contained in ellipsoid (or is empty)
- Good online lecture: https://www.coursera.org/lecture/advanced-algorithms-and-complexity/optional-the-ellipsoid-algorithm-N9rzA

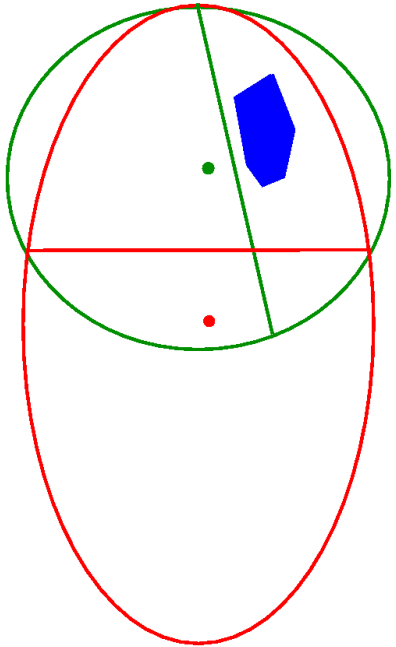


image source: WIkipedia

Karmarkar's algorithm, poly-time with practical impact

[11] **Patent Number: 4,744,028**

[45] **Date of Patent: May 10, 1988**

[54] **METHODS AND APPARATUS FOR EFFICIENT RESOURCE ALLOCATION**

[75] Inventor: **Narendra K. Karmarkar**, Somerset, N.J.

[73] Assignee: **American Telephone and Telegraph Company, AT&T Bell Laboratories**, Murray Hill, N.J.

[21] Appl. No.: **725,342**

[22] Filed: **Apr. 19, 1985**

[51] Int. Cl.[4] G06F 15/20; H04Q 3/66; H04M 7/00

[52] U.S. Cl. 364/402

[58] Field of Search 364/402; 379/113, 221; 340/524

[56] References Cited

U.S. PATENT DOCUMENTS

4,364,115	12/1982	Asai	364/765
4,479,176	10/1984	Grimshaw	369/168
4,481,600	11/1984	Asai	364/765

Karmarkar at Bell Labs: an equation to find a new way through the maze

Folding the Perfect Corner

A young Bell scientist makes a major math breakthrough

Every day 1,200 American Airlines jets crisscross the U.S., Mexico, Canada and the Caribbean, stopping in 110 cities and bearing over 80,000 passengers. More than 4,000 pilots, copilots, flight personnel, maintenance workers and baggage carriers are shuffled among the flights; a total of 3.6 million gal. of high-octane fuel is burned. Nuts, bolts, altimeters, landing gears and the like must be checked at each destination. And while performing these scheduling gymnastics, the company must keep a close eye on costs, projected revenue and profits.

Like American Airlines, thousands of companies must routinely untangle the myriad variables that complicate the efficient distribution of their resources. Solving such monstrous problems requires the use of an abstruse branch of mathematics known as linear programming. It is the kind of math that has frustrated theoreticians for years, and even the fastest and most powerful computers have had great difficulty juggling the bits and pieces of data. Now Narendra Karmarkar, a 28-year-old Indian-born mathematician at Bell Laboratories in Murray Hill, N.J., after only a years' work has cracked the puzzle of linear programming by devising a new algorithm, a step-by-step mathematical formula. He has translated the procedure into a program that should allow computers to track a greater combination of tasks than ever before and in a fraction of the time.

Unlike most advances in theoretical mathematics, Karmarkar's work will have an immediate and major impact on the real world. "Breakthrough is one of the most abused words in science," says Ronald Graham, director of mathematical sciences at Bell Labs. "But this is one situation where it is truly appropriate."

Before the Karmarkar method, linear equations could be solved only in a cumbersome fashion, ironically known as the simplex method, devised by Mathematician George Dantzig in 1947. Problems are conceived of as giant geodesic domes with thousands of sides. Each corner of a facet on the dome

image source: M. Grötschel

Barrier method

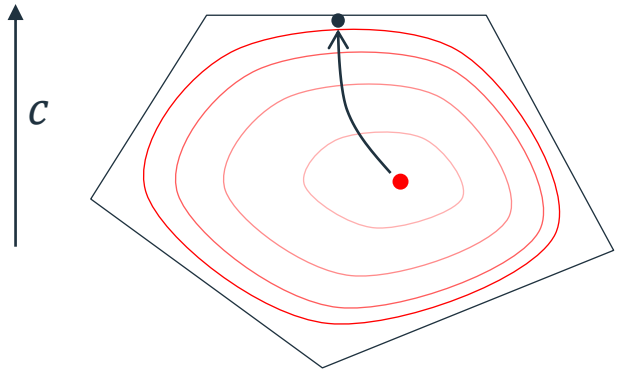
- Instead of $\min \{c^T x \mid Ax = b, x \geq 0\}$ solve $\min \{c^T x - \mu \sum \ln x_i \mid Ax = b, x \geq 0\}$
  - Strictly convex problem, has a single unique solution (when original problem feasible)
- For small x, -ln x becomes large, hence solution is an interior point
  - Converges to optimum of original LP when $\mu \to 0$
- Integrate primal and dual LP into the following nonlinear(!) equation system:
  - $Ax = b$ primal
  - $yA + s = c$ dual
  - $xs = \mu$ complementary slackness
  - $x, s \geq 0$
- This can be solved by a Newton method

Crossover

- Barrier solutions are not basic
  - Typically only few columns at their bound
  - Can neither be used for simplex warmstart, nor for Gomory cuts
  - Numerically slightly off
- Crossover: Creates a basic vertex solution from a nonbasic interior point solution
- Algorithm similar to simplex
- Guesses initial basis (crash) and nonbasis (note: some columns might be at their bounds), maintains set of superbasic columns (not at their bound and not basic) and tries to push those to zero or to push a basic column to zero and a superbasic into the basis
  - Primal and dual crossover
  - Polynomial-time algorithm

Analytic center

- Solve $\min \{- \mu \sum \ln x_i \mid Ax = b, x \geq 0\}$
  - barrier system without original objective
- This computes the analytic center of the polytope
  - Point that maximizes distance to the boundary



- Several MIP applications make use of analytic center:
- In standalone primal heuristics:
  - Recursive central rounding (Naoum-Sawaya 2013)
  - Use AC in Feasibility Pump (Baena and Castro 2011, Boland et al 2011)
  - Metaheuristic based on interior point solutions (Plateau et al 2001)
- In mixed-integer convex optimization:
  - Analytic center cutting plane method (Gondzio, du Merle et al 1996)
  - Also for MIP applications (Ferris et al 2001, Fischetti, Salvagnin 2010)
  - Branching, presolving (Berthold et al 2018)

Quiz time

- An LP (in standard from) with 20 constraints over 100 variables
  a) will have at most 20 nonzeros in an optimal basic solution
  b) might have more than 20, but at most 80 nonzeros in an optimal basic solution
  c) might have more than 80, but at most 100 nonzeros in an optimal basic solution
- We can warm start the Primal Simplex algorithm after
  a) adding a row
  b) changing a variable bound
  c) changing the objective
- The Barrier algorithm converges to
  a) a vertex solution
  b) a point in the center of the optimal face
  c) a point outside the polyhedron

Quiz time

- An LP (in standard from) with 20 constraints over 100 variables
  a) will have at most 20 nonzeros in an optimal basic solution
  b) might have more than 20, but at most 80 nonzeros in an optimal basic solution
  c) might have more than 80, but at most 100 nonzeros in an optimal basic solution
- We can warm start the Primal Simplex algorithm after
  a) adding a row
  b) changing a variable bound
  c) changing the objective
- The Barrier algorithm converges to
  a) a vertex solution
  b) a point in the center of the optimal face
  c) a point outside the polyhedron

Gomory Cuts

Click to add text

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.

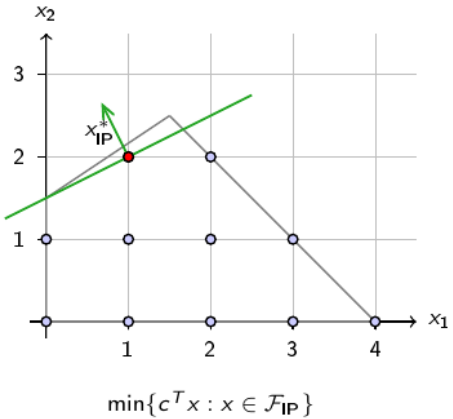
It is the purpose of this note to outline a finite algorithm for obtaining integer solutions to linear programs. The algorithm has been programmed successfully on an E101 computer and used to run off the integer solution to small (seven or less variables) linear programs completely automatically.

image source: ralphgomory.com

Integer Programming

$$\mathcal{F}_{IP} := \{x \in \mathbb{Z}^n_+ : Ax \leq b\}$$

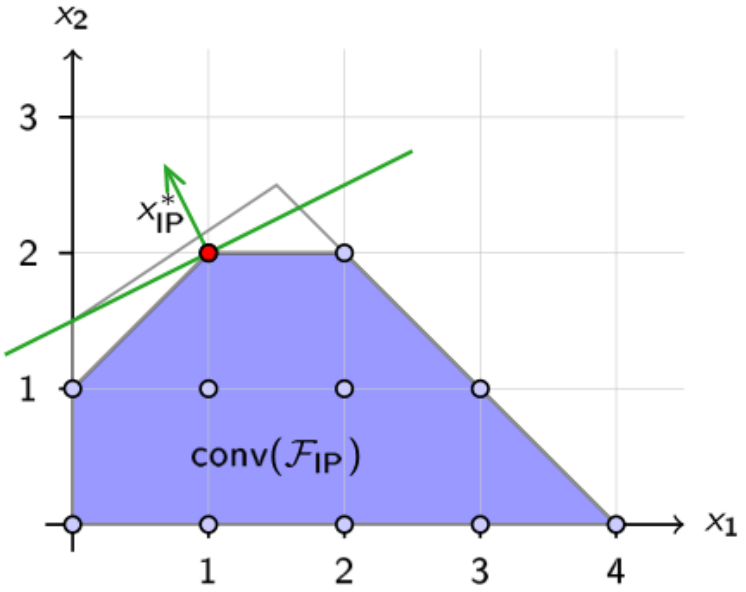
$$\mathcal{F}_{LP} := \{x \in \mathbb{R}^n_+ : Ax \leq b\}$$



- Optimizing over a discrete set
- Optimal solution not necessarily in a vertex

General cutting plane method

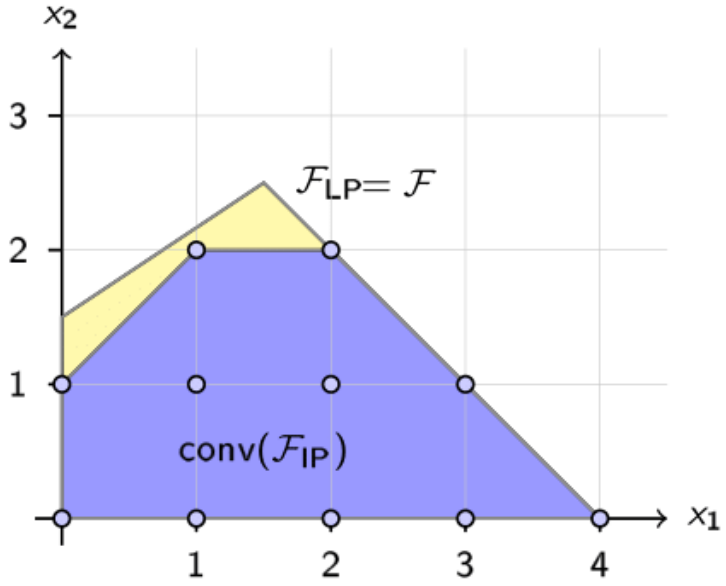
- $conv(F_{IP})$ is a polyhedron
- Principally, every IP could be formulated as LP
- Problems:
  - Linear description not known
  - Might require exponentially many rows
- Motivation:
  - For optimal solution, we need at most n rows



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

General cutting plane method: Colorful picture

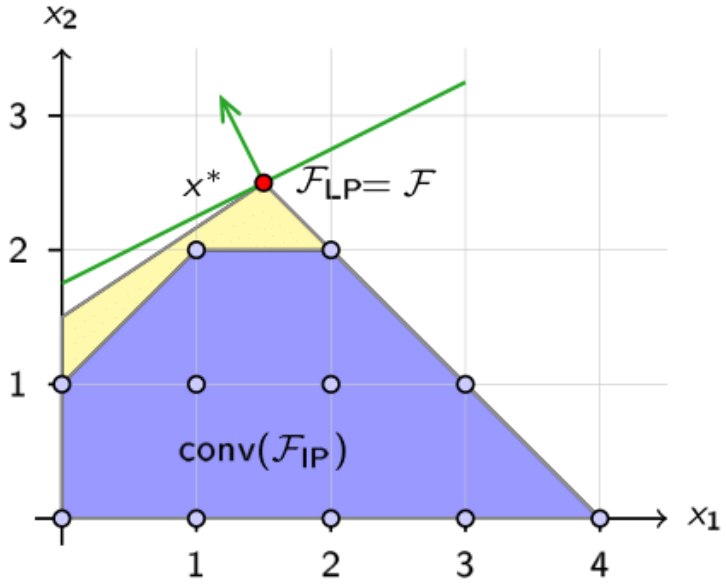
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

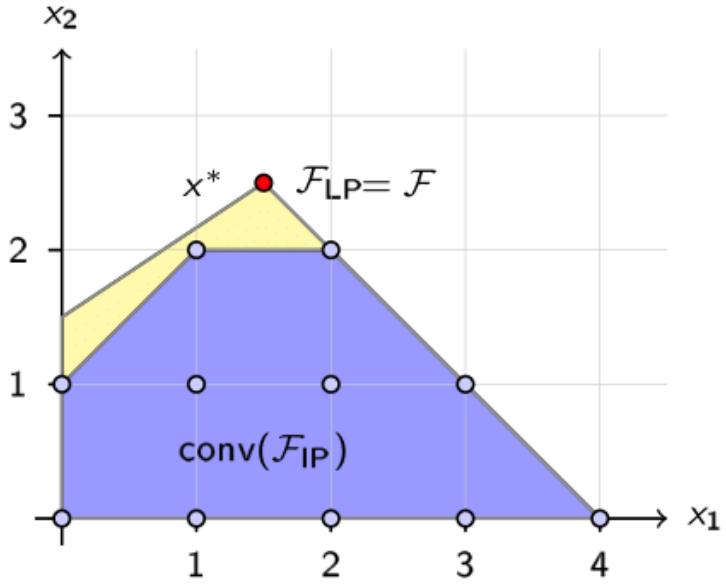
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

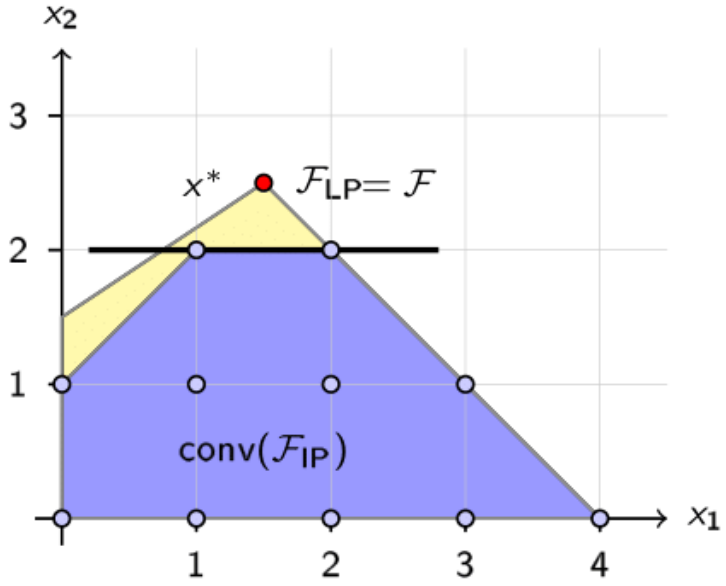
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

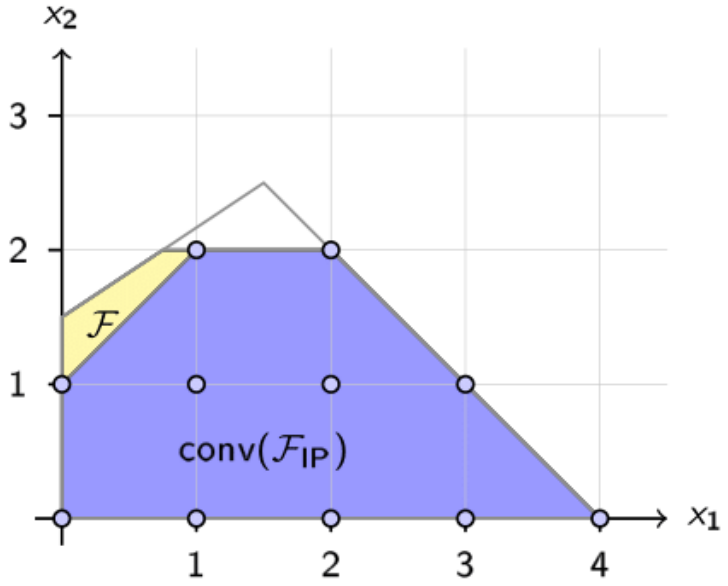
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

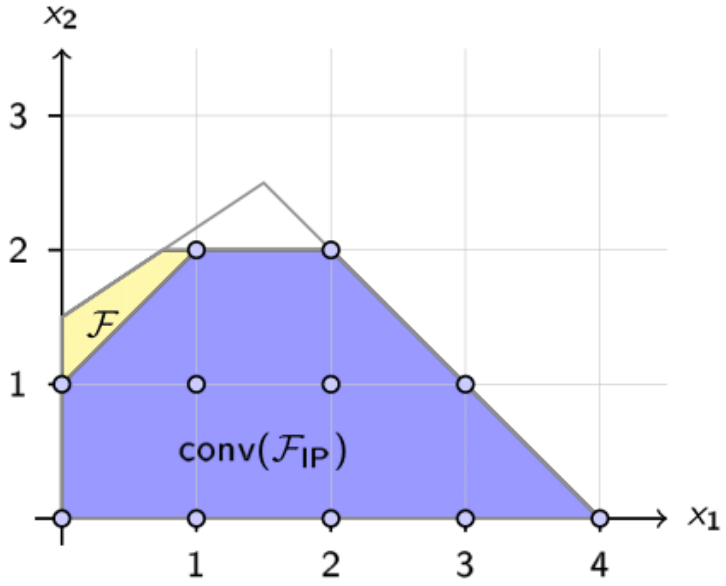
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

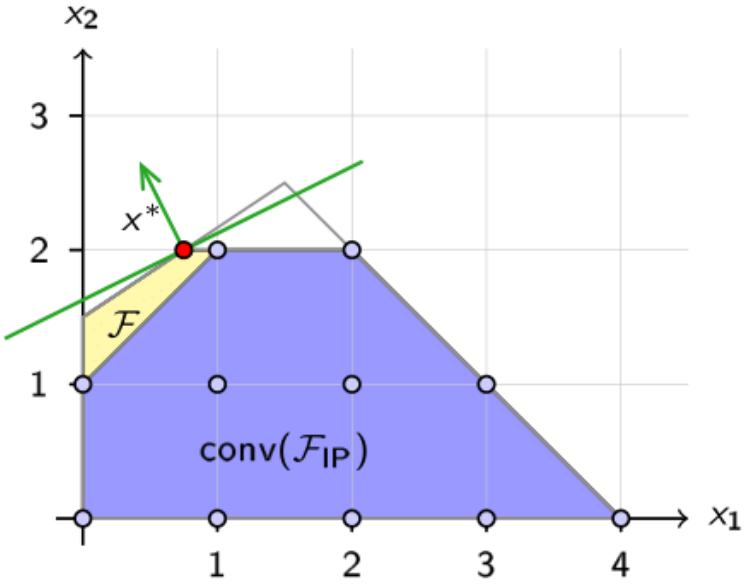
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

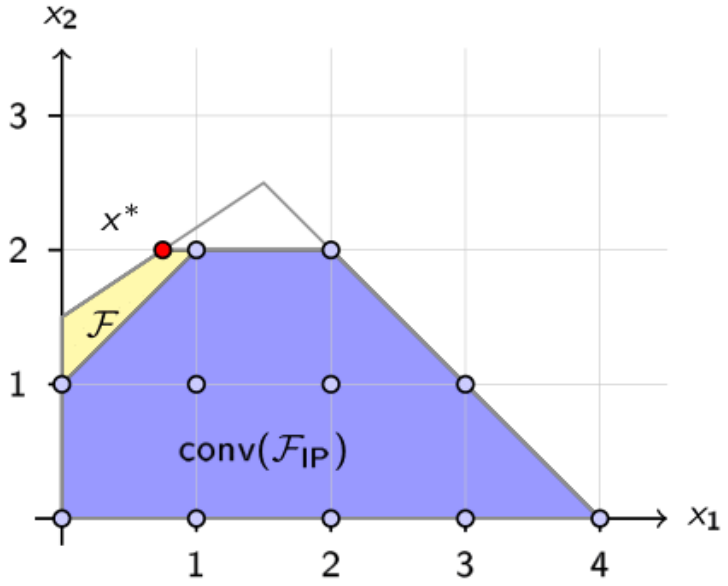
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

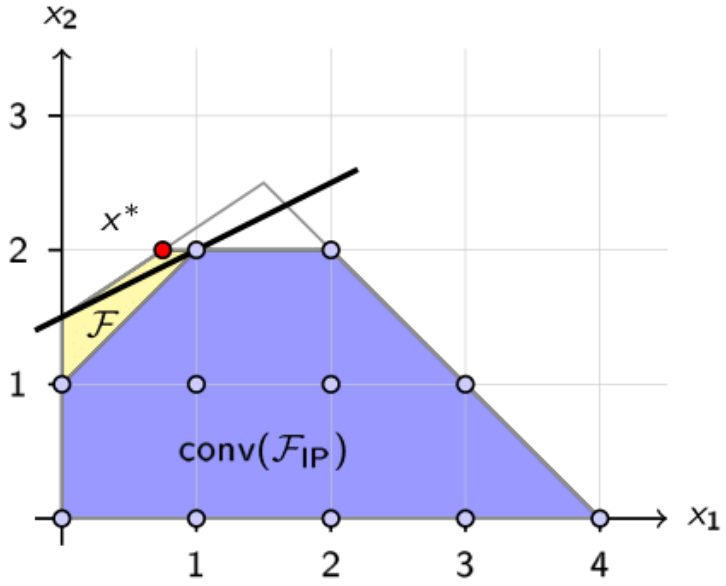
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

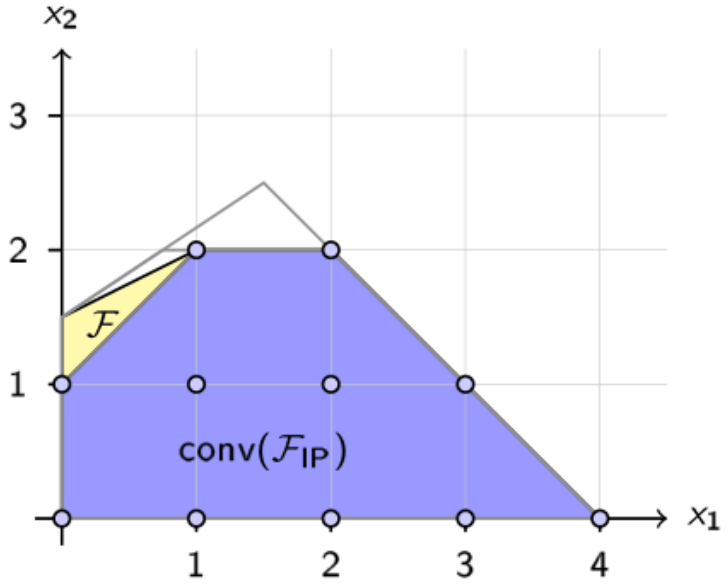
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

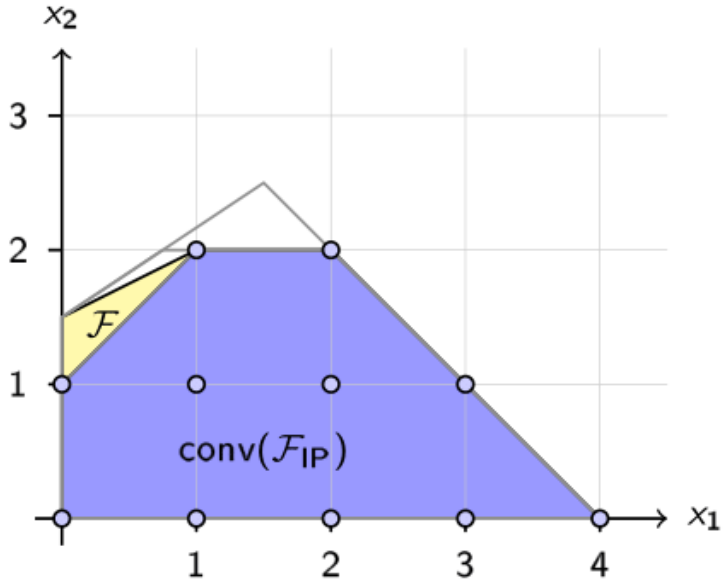
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

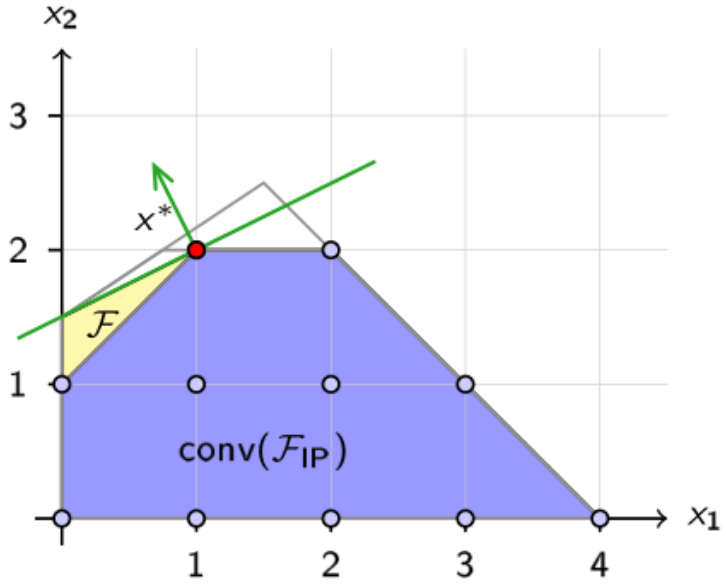
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

General cutting plane method: Colorful picture

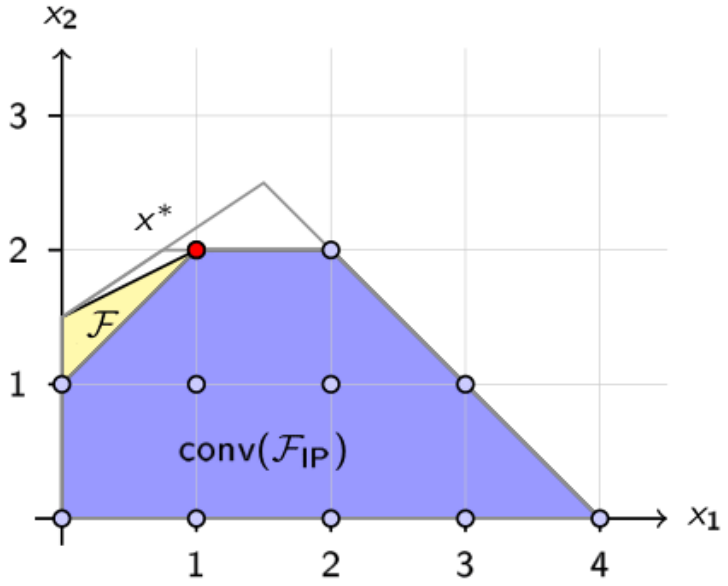
1. Initialize: $F \leftarrow F_{LP}$
2. Solve $x^* \leftarrow \min\{c^T x \mid x \in F\}$
3. If $x^* \in F_{IP}$:
   Stop!
4. Add inequality to F that is:
   - Valid for $conv(F_{IP})$ and
   - Violated by x^*
5. Goto 2.



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

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.



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

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

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
- 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
- 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 is less equal zero

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 is less equal zero
- Hence, $-\sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq \lfloor \bar{b}_i \rfloor - \bar{b}_i$ 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 is less equal zero
- Hence, $-\sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq \lfloor \bar{b}_i \rfloor - \bar{b}_i$ is a valid inequality for the given IP
  - And it is violated by the basic LP solution, since the right hand side is zero, but the right hand 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 is less equal zero
- Hence, $-\sum (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \leq \lfloor \bar{b}_i \rfloor - \bar{b}_i$ is a valid inequality for the given IP
  - And it is violated by the basic LP solution, since the right hand side is zero, but the right hand 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)

Gamification

- By Gonzalo Muñoz: http://cuttingplanegame.gonzalomunoz.org/
- Get yourself on the leaderboard and win a prize! (code: cutsatwork)

Branch&Bound

Click to add text

First IFORS meeting: kickstart for LP-based Branch&Bound?

- From Bill Cook's fantastic IFORS distinguished lecture: https://www.youtube.com/watch?v=5VjphFYQKj8

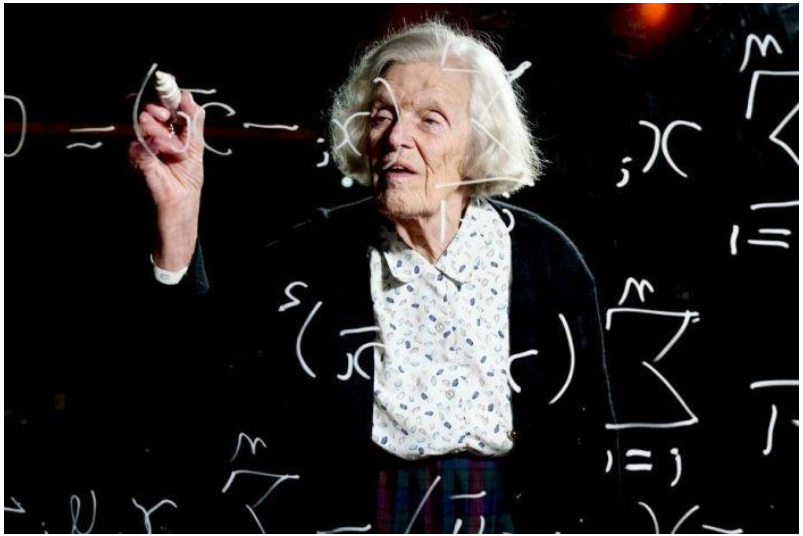


image source: abc.net.au

- Alison Harcourt recently appointed Senior Australian of the year 2019

Branch&Bound

- Divide&Conquer
- Split search region into strictly smaller subproblems, hopefully easier to solve, iterate
  - Typically two subproblems, typically disjoint
- Maintain lower and upper bounds, prune subproblems outside of bounds
- General version for discrete problems by Land&Doig 1960
- Easier implementation, MIP-specific, with LP-relaxations, by Dakin 1965
  - Dual simplex shines: Warm starts when adding new variable bounds

LP-based Branch&Bound: Algorithm

Steps

1. Abort criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility check
6. Branching

$\mathcal{L} \leftarrow \{P\}, \quad U \leftarrow \infty, \quad x^{IP} \leftarrow$ NULL;

if $\mathcal{L} = \emptyset$ **then return** x^{IP} *and* U;

Select $P_i \in \mathcal{L}, \quad \mathcal{L} \leftarrow \mathcal{L} \backslash \{P_i\}$;

Solve LP relaxation of P_i, $\quad L_{\text{loc}} \leftarrow c(x^{\text{LP}})$ or ∞;

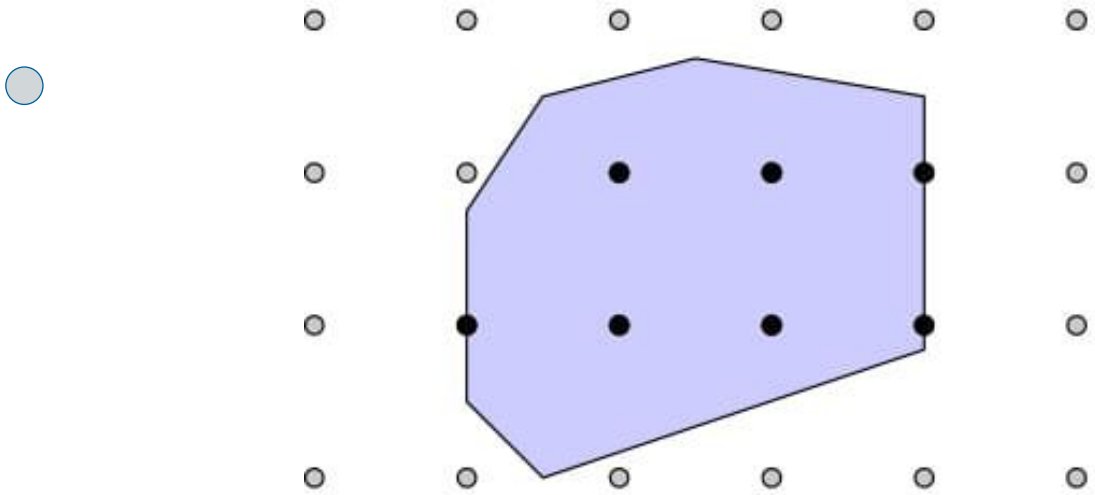
if $L_{loc} \geq U$ **then goto** line 2;

if $x^{LP} \in P$ **then** $x^{\text{IP}} \leftarrow x^{\text{LP}}, \quad U \leftarrow L_{\text{loc}}, \quad$ **goto** line 2;

Select $j \in I : x_j^{\text{LP}} \notin \mathbb{Z}$. Split P_i into $P_{2i+1} := P_i \cup \{x_j \leq \lfloor x_j^{\text{LP}} \rfloor\}$, and $P_{2i+2} := P_i \cup \{x_j \geq \lceil x_j^{\text{LP}} \rceil\}, \quad \mathcal{L} \leftarrow \mathcal{L} \cup \{P_{2i+1}, P_{2i+2}\}, \quad$ **goto** line 2;

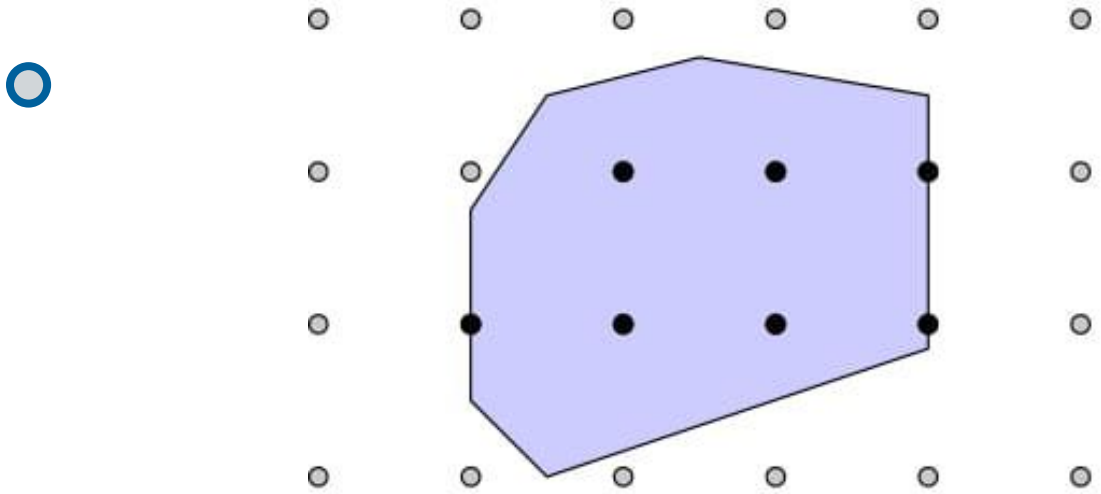
LP-based Branch&Bound (colorful picture)

1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching



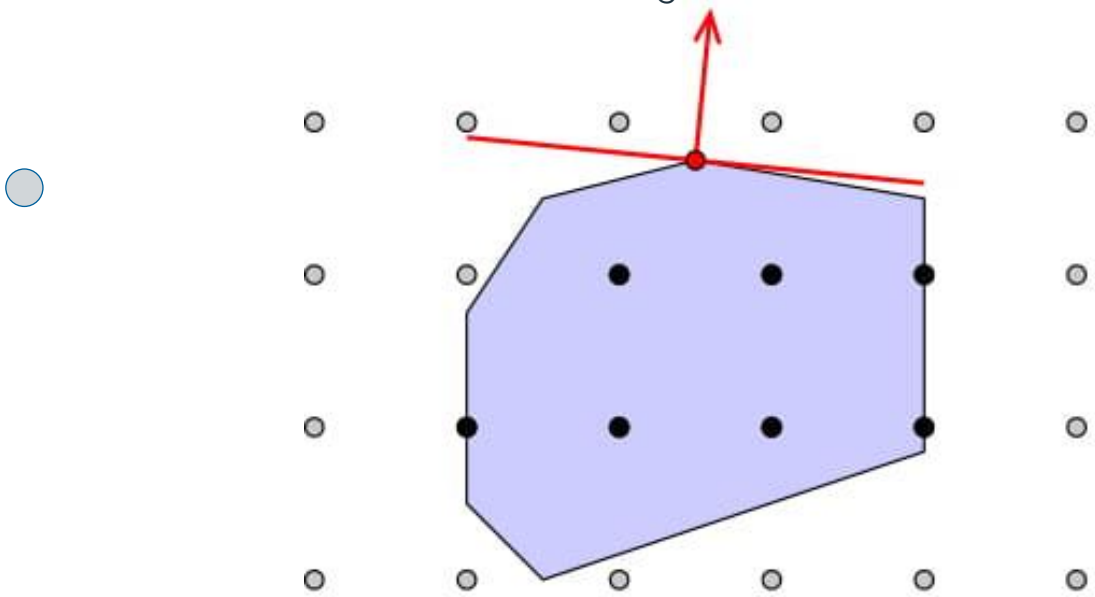
LP-based Branch&Bound (colorful picture)

1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching



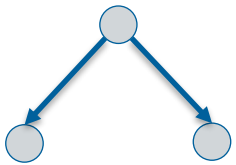
LP-based Branch&Bound (colorful picture)

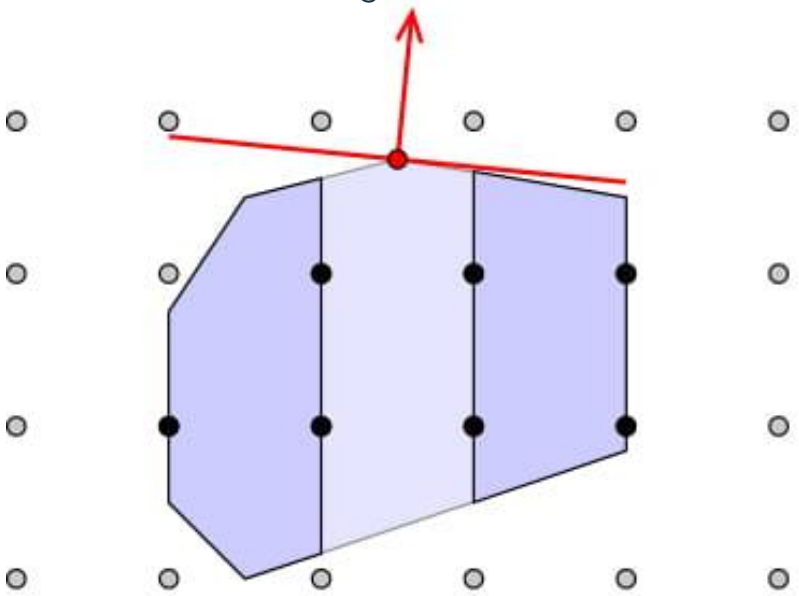
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching



LP-based Branch&Bound (colorful picture)

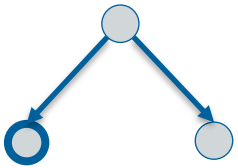
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

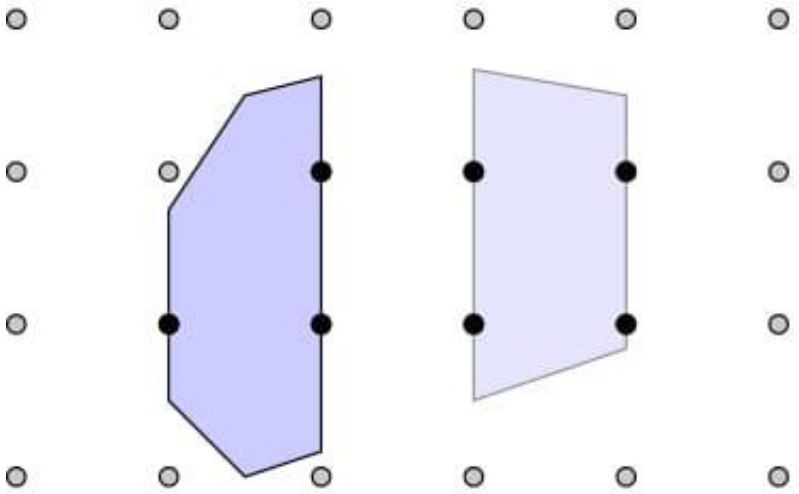




LP-based Branch&Bound (colorful picture)

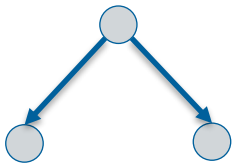
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

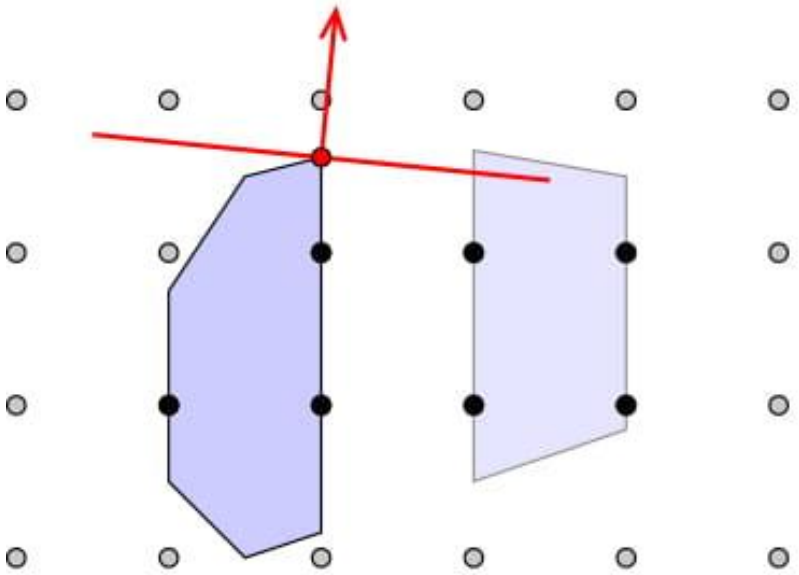




LP-based Branch&Bound (colorful picture)

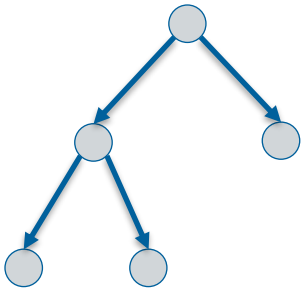
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

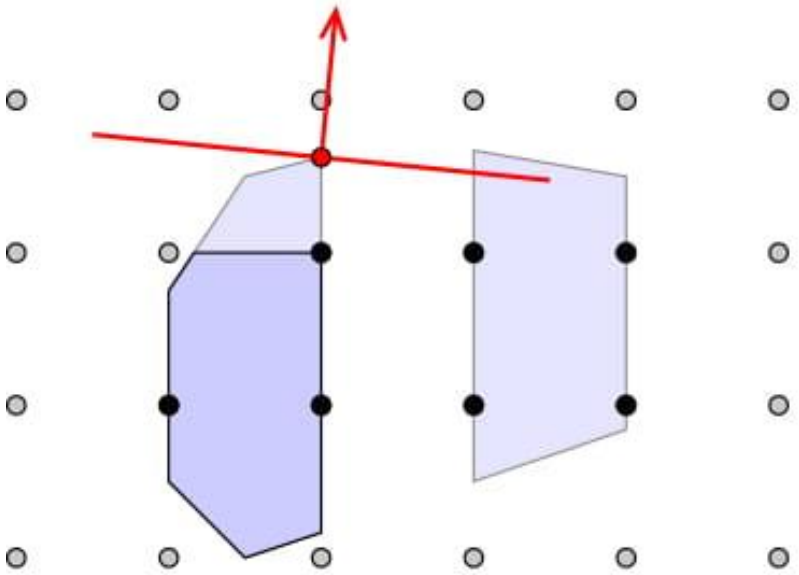




LP-based Branch&Bound (colorful picture)

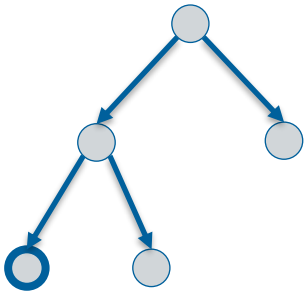
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

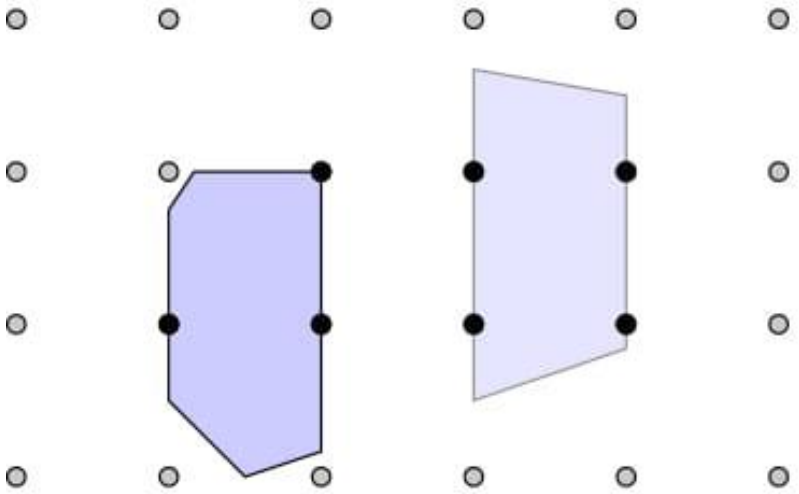




LP-based Branch&Bound (colorful picture)

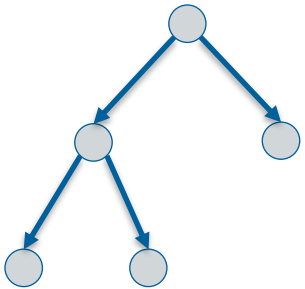
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

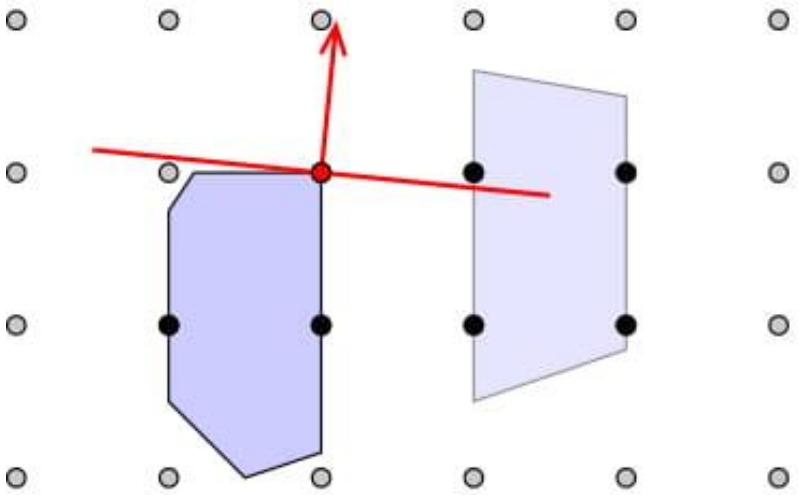




LP-based Branch&Bound (colorful picture)

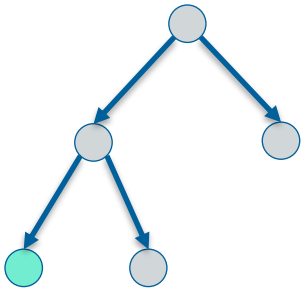
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

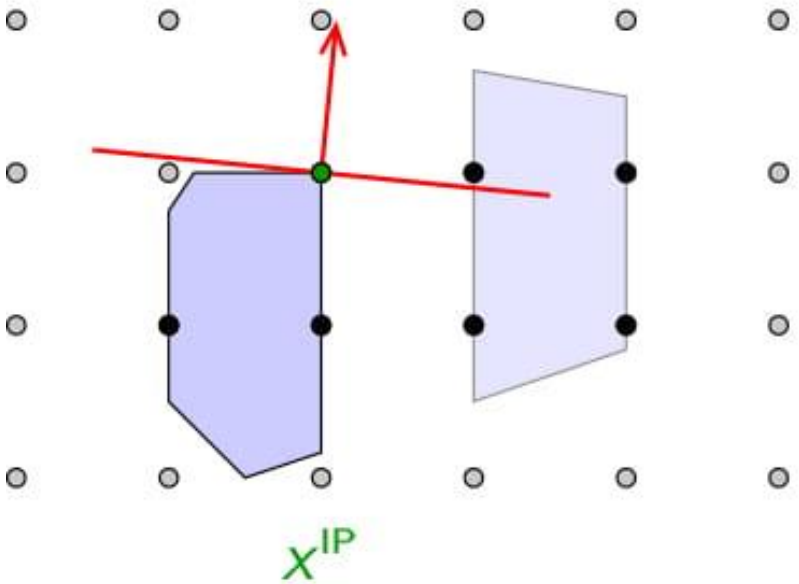




LP-based Branch&Bound (colorful picture)

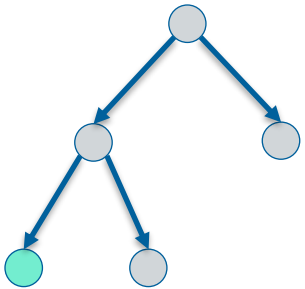
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

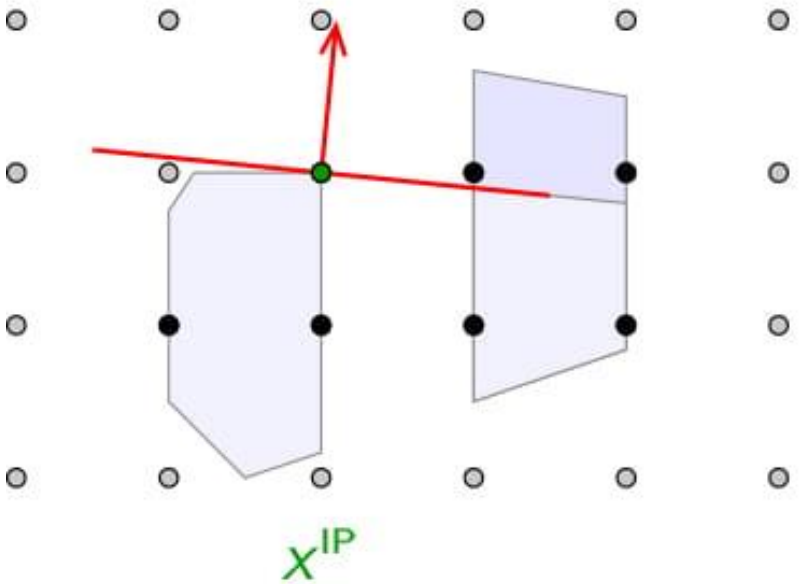




LP-based Branch&Bound (colorful picture)

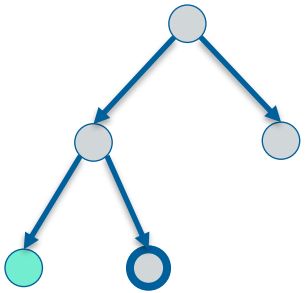
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

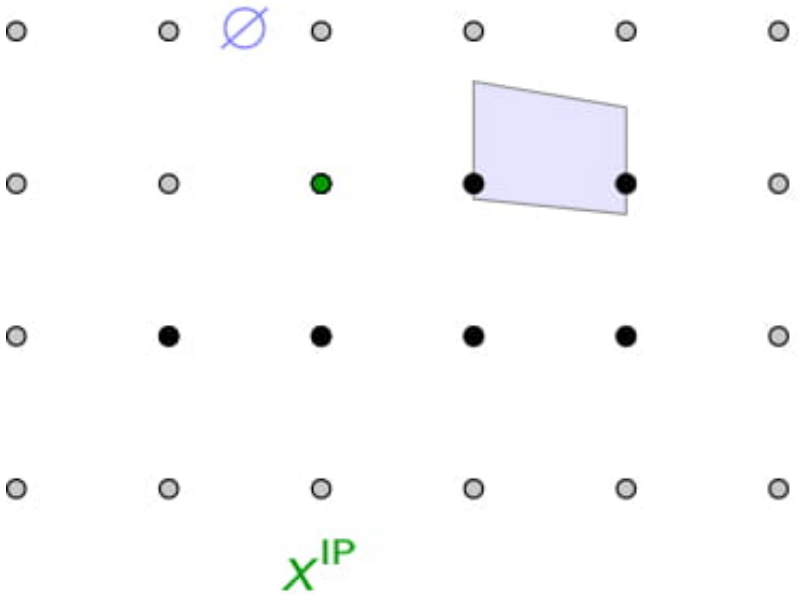




LP-based Branch&Bound (colorful picture)

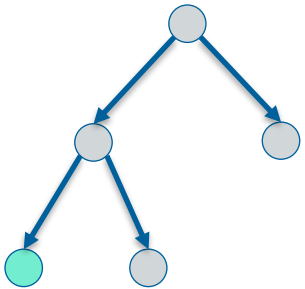
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

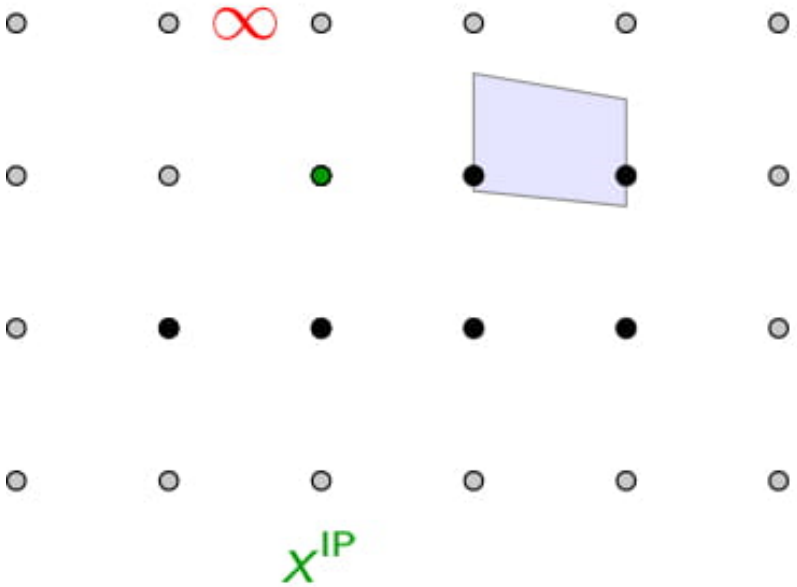




LP-based Branch&Bound (colorful picture)

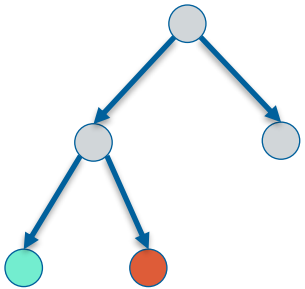
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

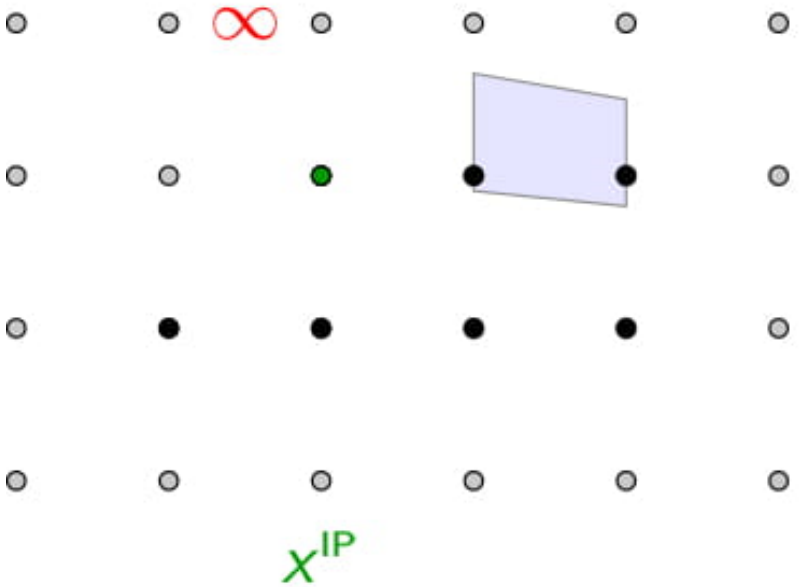




LP-based Branch&Bound (colorful picture)

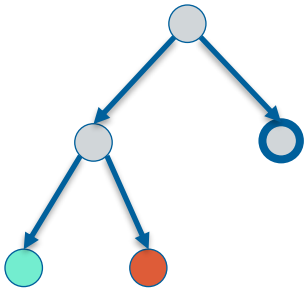
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

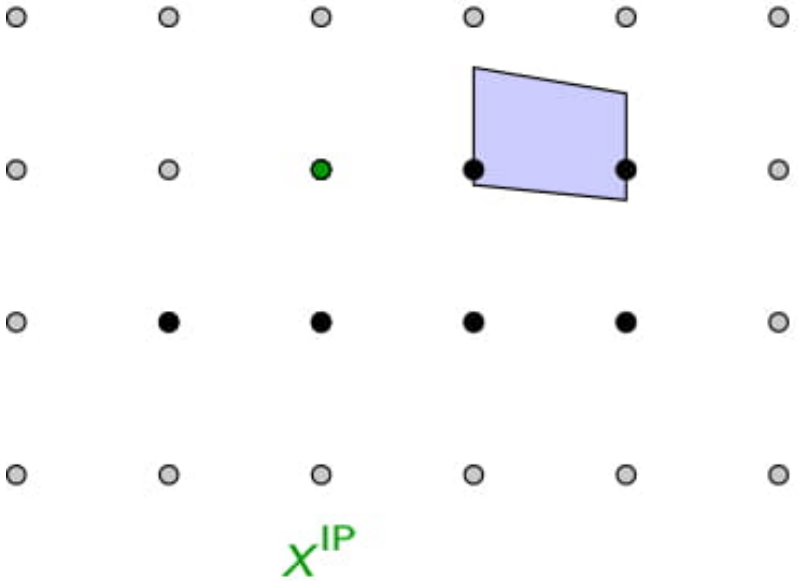




LP-based Branch&Bound (colorful picture)

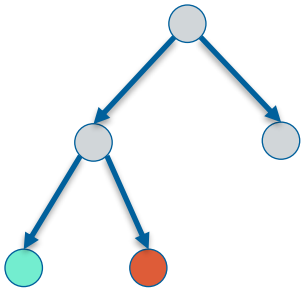
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

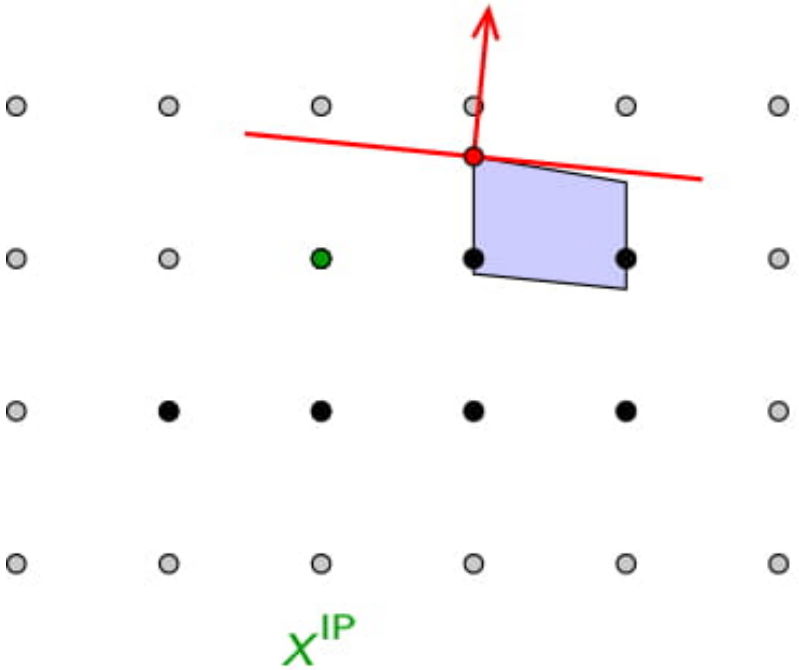




LP-based Branch&Bound (colorful picture)

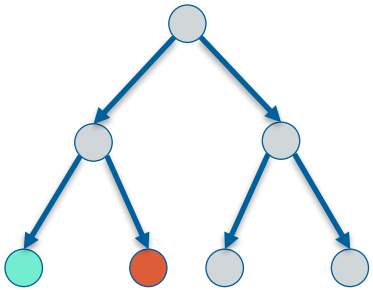
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

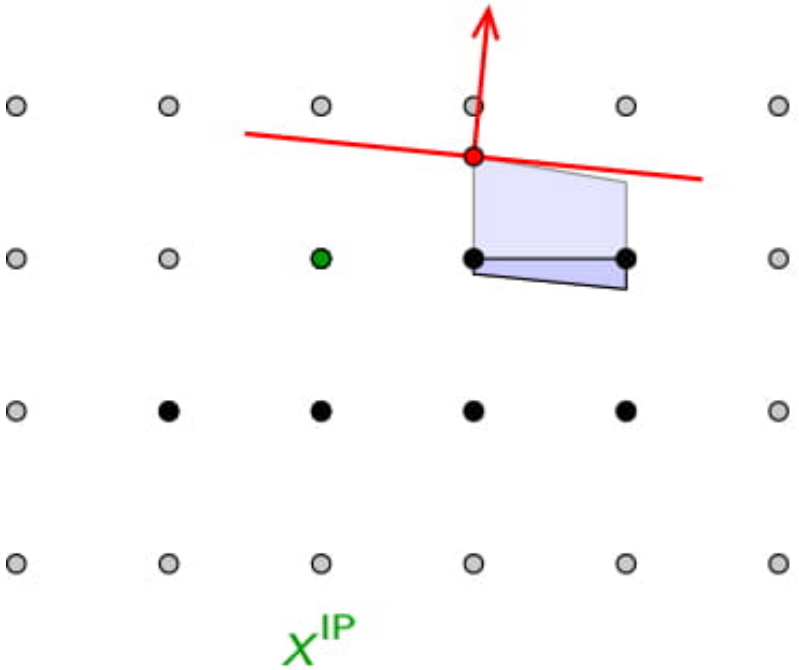




LP-based Branch&Bound (colorful picture)

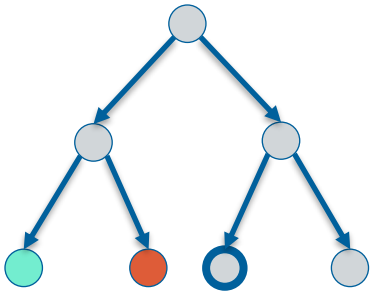
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

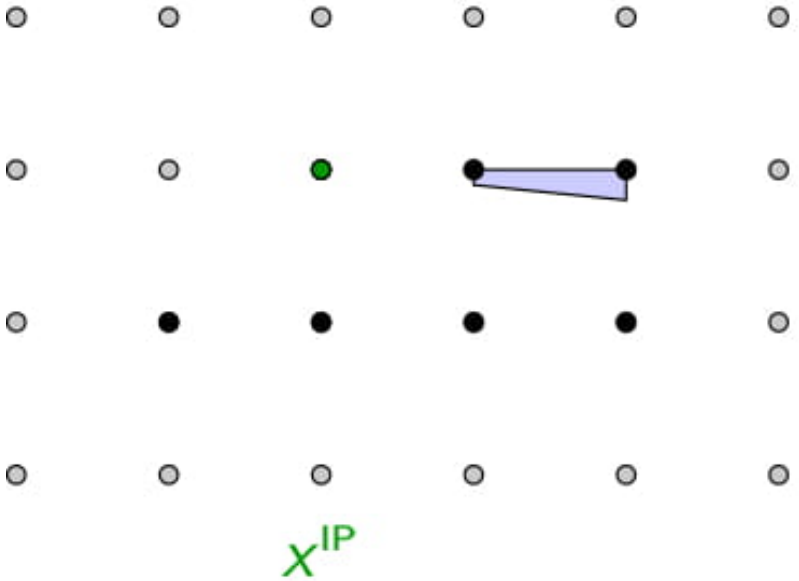




LP-based Branch&Bound (colorful picture)

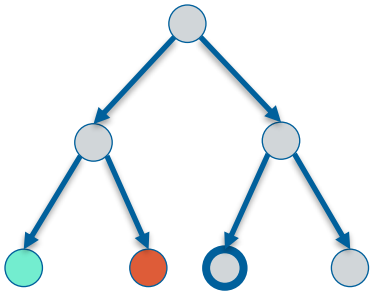
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

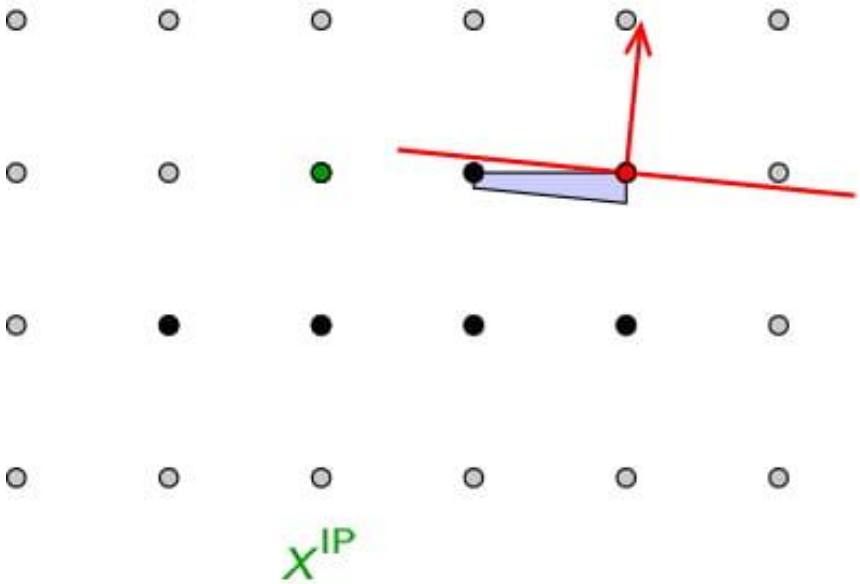




LP-based Branch&Bound (colorful picture)

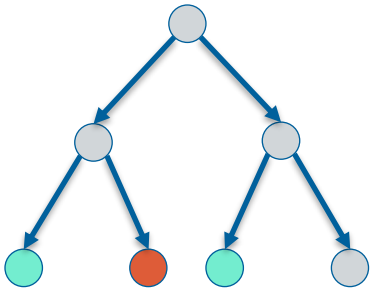
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

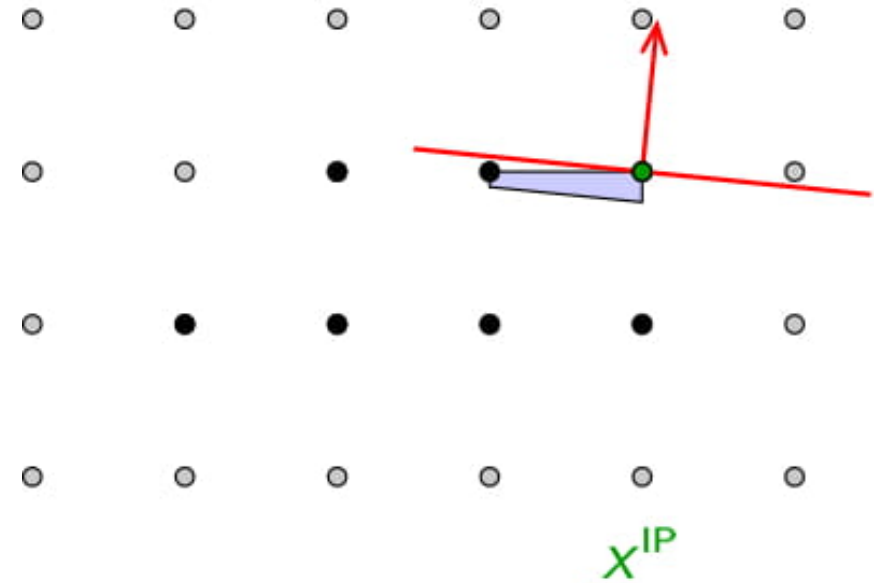




LP-based Branch&Bound (colorful picture)

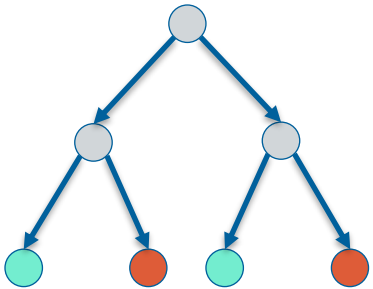
1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching

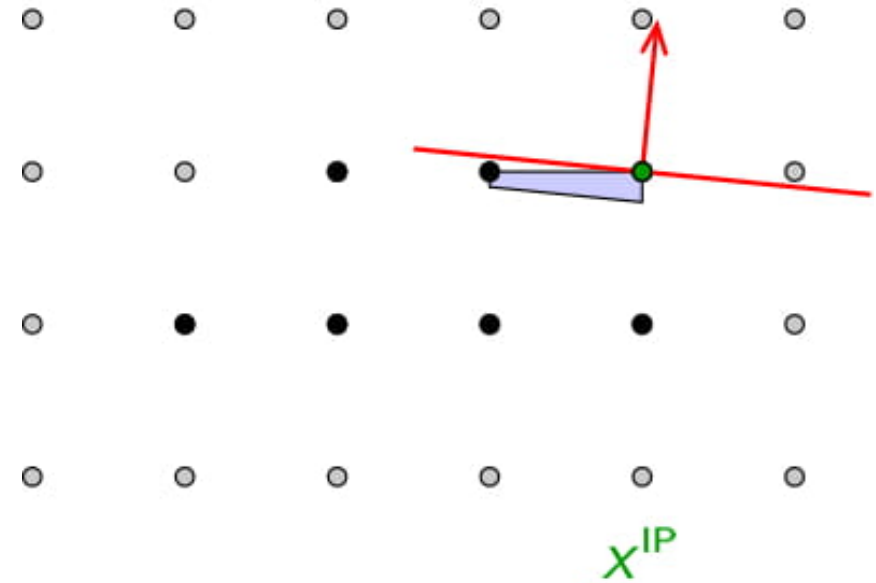




LP-based Branch&Bound (colorful picture)

1. Abort Criterion
2. Node selection
3. Solve relaxation
4. Bounding
5. Feasibility Check
6. Branching





Running time

- Can obviously be exponential, but is it even finite?

```
Minimize
  obj: x3
Subject To
  c1: 12 x1 + 9 x2 - x3 = 0
Bounds
  -inf <= x1
  -inf <= x2
  1 <= x3
General
  x1 x2 x3
End
```

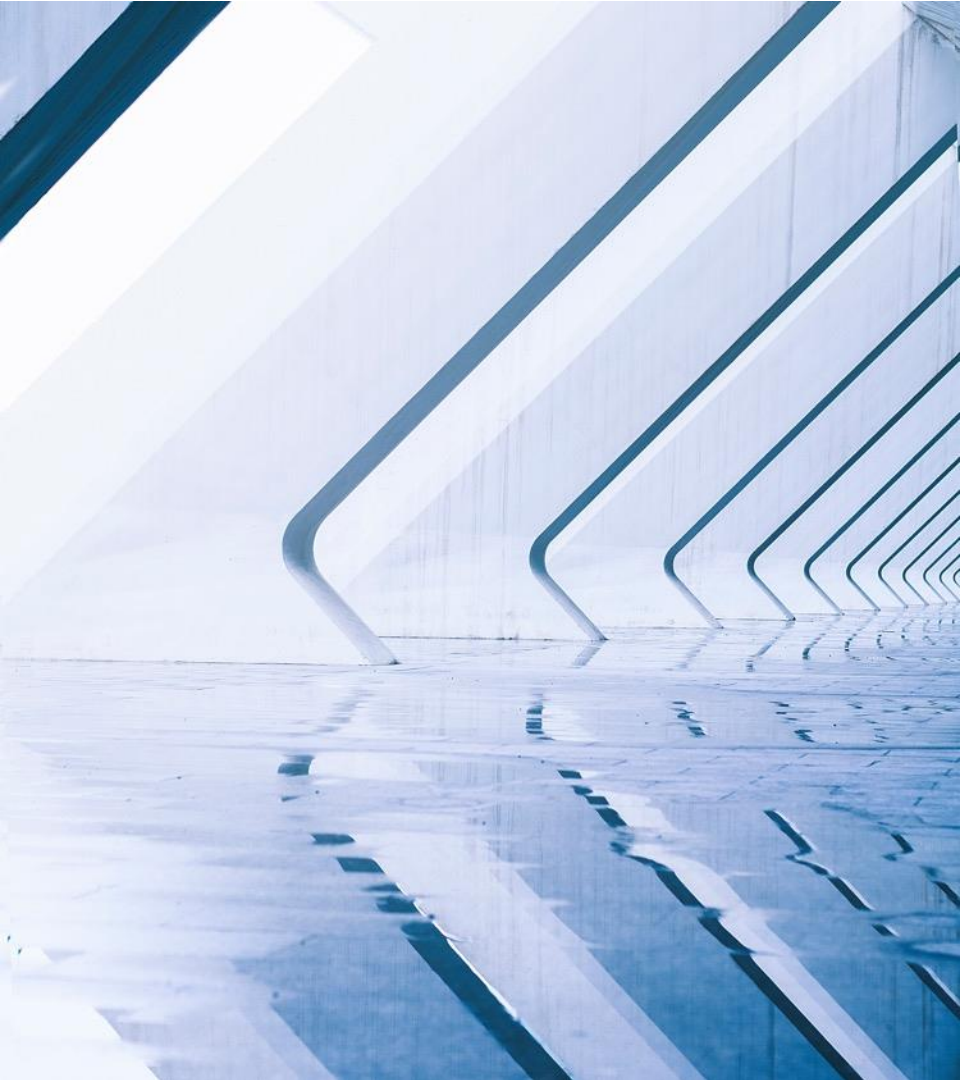
Careful modelling matters!

Quiz time

- With Gomory's algorithm, the first integer solution found
  a) will be an optimal solution
  b) might be suboptimal, but within a factor 2 of optimality
  c) has a minimum number of nonzero variables
- You get one Gomory cut per
  a) basic solution
  b) basic variable
  c) fractional variable
- The LP algorithm of choice for LP-based branch-and-bound is:
  a) Primal Simplex
  b) Dual Simplex
  c) Barrier

Quiz time

- With Gomory's algorithm, the first integer solution found
  a) will be an optimal solution
  b) might be suboptimal, but within a factor 2 of optimality
  c) has a minimum number of nonzero variables
- You get one Gomory cut per
  a) Basic solution
  b) Basic variable
  c) Fractional variable
- The LP algorithm of choice for LP-based branch-and-bound is
  a) Primal Simplex
  b) Dual Simplex
  c) Barrier



FICO

Thank You!

Timo Berthold