Solving Mixed Integer Linear Programs with Cutting Planes

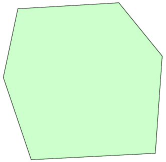
Tobias Kohler

May 16, 2024

Mixed Integer Linear Program

- Optimize linear objective function s.t. linear constraints and some integer constraints.
- Sometimes, decision variables are discrete: Distribution of patients/supplies/vehicles... or binary/logical variables.

Mixed Integer Linear Program and Linear Program Relaxation



MILP (standard form)

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & x \in F \\ & := \{x \mid Ax \leq b,\, x \in \mathbb{Z}^{n_1}_{\geq 0} \times \mathbb{R}^{n-n_1}_{\geq 0}\} \end{aligned}$$

$c, x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

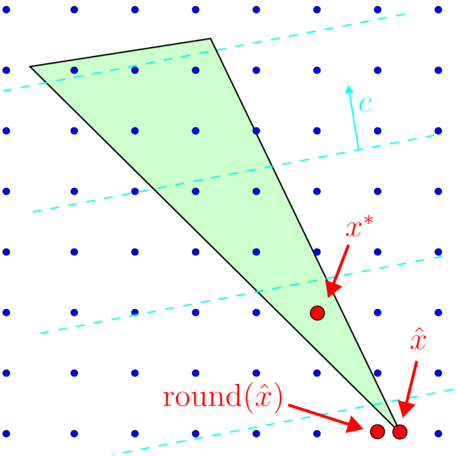
LP Relaxation

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & x \in P \\ & := \{x \mid Ax \leq b,\, x \in \mathbb{R}^{n}_{\geq 0}\} \end{aligned}$$

Notation

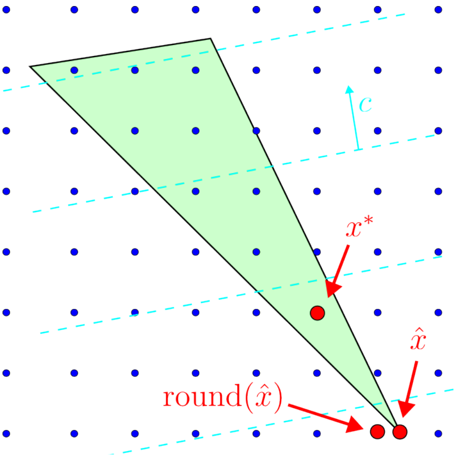
- $x_1, \ldots, x_{n_1}$: Integer variables
- $x_{n_1}, \ldots, x_n$: Real variables
- F, x^* : Feasible region and optimal solution of the MILP
- P, $\hat{x}$: Feasible region and optimal solution of the LP-relaxation $(F = P \cap \mathbb{Z}^{n_1}_{\geq 0} \times \mathbb{R}^{n-n_1}_{\geq 0})$
- $n_1 = 0 \Rightarrow$ Linear Program (LP)
- $n_1 = n \Rightarrow$ Integer Linear Program (ILP)
- $x_i \in \{0, 1\} \Rightarrow$ (Mixed) Binary Program ((M)BP)

Observations



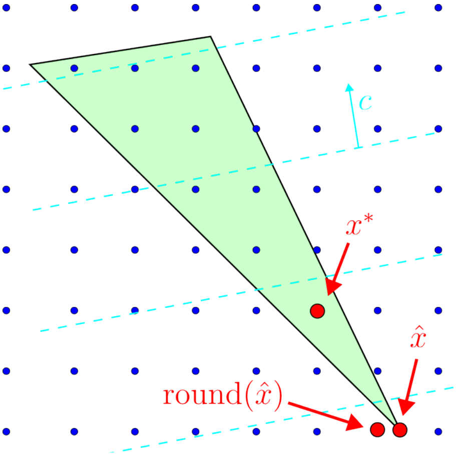
- $\hat{x}$ can be found at a vertex of P (Simplex Algorithm).

Observations



- $\hat{x}$ can be found at a vertex of P (Simplex Algorithm).
- In general, $\text{round}(\hat{x}) \neq x^*$

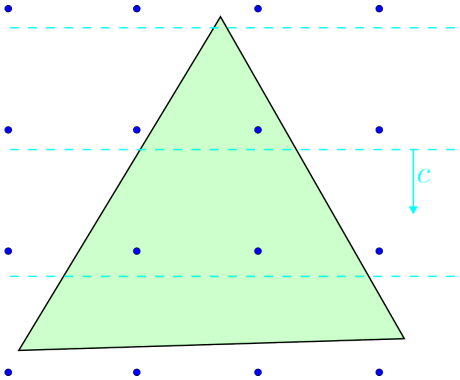
Observations



- $\hat{x}$ can be found at a vertex of P (Simplex Algorithm).
- In general, round($\hat{x}$) $\neq x^*$
- If $\hat{x}$ is already feasible, then $c^\top \hat{x} = c^\top x^*$.

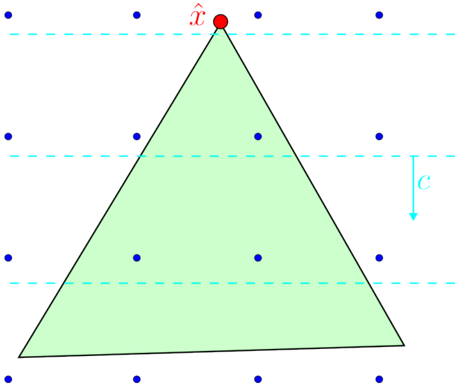
Cutting Planes

Solve the problem relaxation. If integer constraints are violated, add additional inequalities to the problem that cut off the relaxed solution.



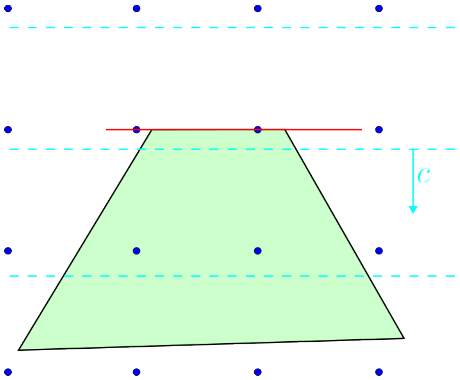
Cutting Planes

Solve the problem relaxation. If integer constraints are violated, add additional inequalities to the problem that cut off the relaxed solution.



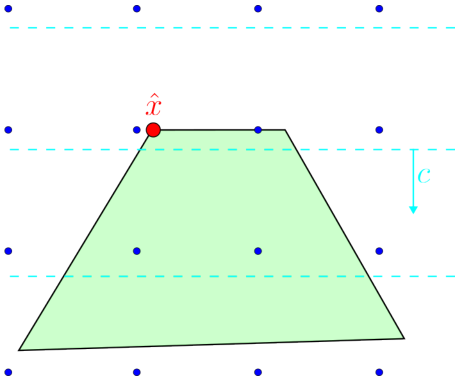
Cutting Planes

Solve the problem relaxation. If integer constraints are violated, add additional inequalities to the problem that cut off the relaxed solution.



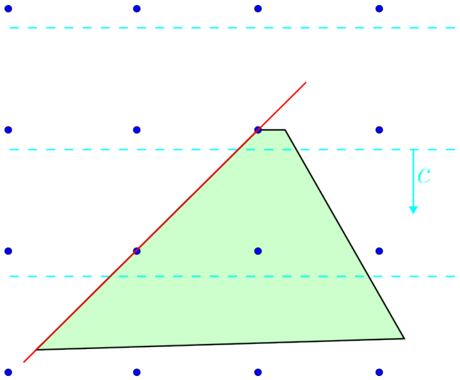
Cutting Planes

Solve the problem relaxation. If integer constraints are violated, add additional inequalities to the problem that cut off the relaxed solution.



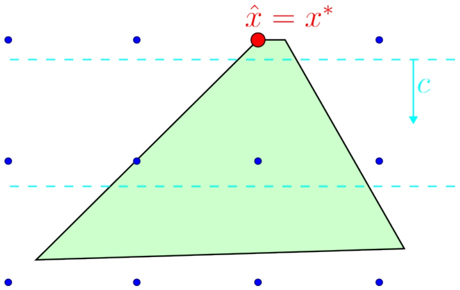
Cutting Planes

Solve the problem relaxation. If integer constraints are violated, add additional inequalities to the problem that cut off the relaxed solution.



Cutting Planes

Solve the problem relaxation. If integer constraints are violated, add additional inequalities to the problem that cut off the relaxed solution.



Valid Inequalities and Cuts

- An inequality $a^\top x \leq r$ is valid for a set F if $a^\top x \leq r$ is satisfied for all $x \in F$.

Valid Inequalities and Cuts

- An inequality $a^\top x \leq r$ is valid for a set F if $a^\top x \leq r$ is satisfied for all $x \in F$.
  - For example: $x \leq 2$ is a valid inequality for $\{x \in \mathbb{Z}_{\geq 0} \mid x \leq 2.718\}$

Valid Inequalities and Cuts

- An inequality $a^\top x \leq r$ is valid for a set F if $a^\top x \leq r$ is satisfied for all $x \in F$.
  - For example: $x \leq 2$ is a valid inequality for $\{x \in \mathbb{Z}_{\geq 0} \mid x \leq 2.718\}$
- A cutting plane (or cut) w.r.t. $\hat{x} \in P \setminus F$ is any valid inequality $a^\top x \leq r$ for F such that:

$$a^\top \hat{x} > r$$

Cutting Planes Algorithm

LP $\leftarrow$ Relaxation of the MILP
repeat
    $\hat{x} \leftarrow$ Optimal solution of the LP
    If the LP is infeasible, return (null, INFEASIBLE)
    If the LP is unbounded and no integer constraints are violated,
    return ($\hat{x}$, UNBOUNDED)
    if $(\hat{x}_1, \ldots, \hat{x}_{n_1}) \notin \mathbb{Z}^{n_1}$ **then**
        Add a cut w.r.t. $\hat{x}$ to the LP
until $(\hat{x}_1, \ldots, \hat{x}_{n_1}) \in \mathbb{Z}^{n_1}$
return ($\hat{x}$, OPTIMAL)

Cutting Strategy

Question: How to generate "good" and useful cuts?

Cutting Strategy

Question: How to generate “good” and useful cuts?

- Good: Cut away as much as possible (while staying feasible)

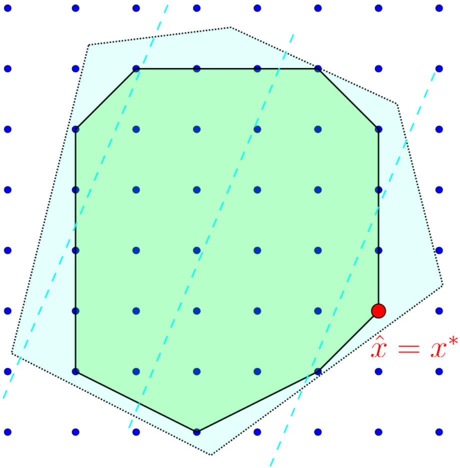
Cutting Strategy

Question: How to generate “good” and useful cuts?

- Good: Cut away as much as possible (while staying feasible)
- Useful: Cut away the optimal solution of the relaxation

Convex Hull

- The relaxed solution $\hat{x}$ in conv(F) also solves the MILP.
- But computing the convex hull is infeasible.
- Our goal is instead to approximate the convex hull in a neighborhood of x^*.



Chvátal–Gomory Inequality for Integer Linear Programs

Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ for an Integer Linear Program ($x \in \mathbb{Z}^n_{\geq 0}$). Then the following inequalities are valid for any $\alpha \geq 0$:

1. $\sum_{j=1}^{n} \alpha a_{ij}x_j \leq \alpha b_i$ $\qquad \alpha \geq 0$

Chvátal–Gomory Inequality for Integer Linear Programs

Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ for an Integer Linear Program ($x \in \mathbb{Z}^n_{\geq 0}$). Then the following inequalities are valid for any $\alpha \geq 0$:

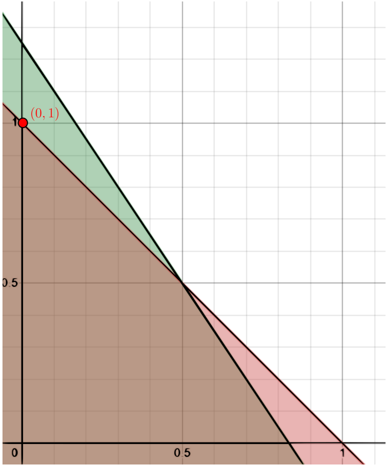
1. $\sum_{j=1}^{n} \alpha a_{ij}x_j \leq \alpha b_i$ $\qquad \alpha \geq 0$
2. $\sum_{j=1}^{n} \lfloor \alpha a_{ij} \rfloor x_j \leq \alpha b_i$ $\qquad x_j \geq 0$

Chvátal–Gomory Inequality for Integer Linear Programs

Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ for an Integer Linear Program ($x \in \mathbb{Z}^n_{\geq 0}$). Then the following inequalities are valid for any $\alpha \geq 0$:

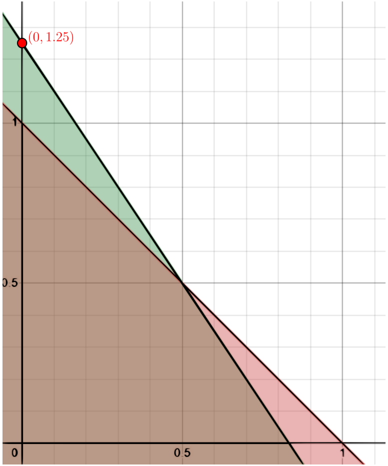
1. $\sum_{j=1}^{n} \alpha a_{ij}x_j \leq \alpha b_i$ $\qquad \alpha \geq 0$
2. $\sum_{j=1}^{n} \lfloor \alpha a_{ij} \rfloor x_j \leq \alpha b_i$ $\qquad x_j \geq 0$
3. $\sum_{j=1}^{n} \lfloor \alpha a_{ij} \rfloor x_j \leq \lfloor \alpha b_i \rfloor$ $\qquad x_j \in \mathbb{Z}$

ILP vs. MILP



- $\min_{x,\,y} -y$
  s.t. $\frac{3}{2}x + y \leq \frac{5}{4}$, $(x,\, y) \in \mathbb{Z}^2_{\geq 0}$
- $(x^*,\, y^*) = (0,\, 1)$
- $\lfloor \frac{3}{2} \rfloor \cdot 0 + \lfloor 1 \rfloor \cdot 1 = 1 \leq \lfloor \frac{5}{4} \rfloor$ ✓

ILP vs. MILP



- $\min_{x,y} -y$
  s.t. $\frac{3}{2}x + y \leq \frac{5}{4}$, $(x,\ y) \in \mathbb{Z}_{\geq 0} \times \mathbb{R}_{\geq 0}$
- $(x^*,\ y^*) = (0,\ \frac{5}{4})$
- $\lfloor \frac{3}{2} \rfloor \cdot 0 + \lfloor 1 \rfloor \cdot \frac{5}{4} = \frac{5}{4} \not\leq \lfloor \frac{5}{4} \rfloor$ ✗

Basic Mixed Integer Rounding Inequalities I

Let $x \in \mathbb{Z}_{\geq 0}$, $y \in \mathbb{R}_{\geq 0}$, $b \in \mathbb{R}_{>0} \setminus \mathbb{Z}$. Then

$$x \leq \lfloor b \rfloor \text{ is a valid inequality for } \{x + y \leq b\} \tag{1}$$

and

$$x \geq \lceil b \rceil \text{ is a valid inequality for } \{-x + y \leq -b\} \tag{2}$$

Basic Mixed Integer Rounding Inequalities I

Let $x \in \mathbb{Z}_{\geq 0}$, $y \in \mathbb{R}_{\geq 0}$, $b \in \mathbb{R}_{>0} \setminus \mathbb{Z}$. Then $x \leq \lfloor b \rfloor$ is a valid inequality for $\{x + y \leq b\}$

Basic Mixed Integer Rounding Inequalities II

Let $x \in \mathbb{Z}_{\geq 0},\ y \in \mathbb{R}_{\geq 0},\ b \in \mathbb{R}_{>0} \setminus \mathbb{Z}$. Then

$$x - \frac{1}{1 - f_b} y \leq \lfloor b \rfloor \text{ is a valid inequality for } \{x - y \leq b\} \tag{1}$$

and

$$x + \frac{1}{f_b} y \geq \lceil b \rceil \text{ is a valid inequality for } \{-x - y \leq -b\} \tag{2}$$

Basic Mixed Integer Rounding Inequalities II

Let $x \in \mathbb{Z}_{\geq 0}$, $y \in \mathbb{R}_{\geq 0}$, $b \in \mathbb{R}_{>0} \setminus \mathbb{Z}$. Then $x - \frac{1}{1-f_b} y \leq \lfloor b \rfloor$ is a valid inequality for $\{x - y \leq b\}$

General Mixed Integer Rounding Inequality

Let $F_{MIR} = \{(x, y) \in \mathbb{Z}^2_{\geq 0} \times \mathbb{R}_{\geq 0} \mid a_1x_1 + a_2x_2 - y \leq b\}$ where $a \in \mathbb{R}^2$, $b \in \mathbb{R} \setminus \mathbb{Z}$ and assume that $f_1 \leq f_b < f_2$. Then the inequality

$$\lfloor a_1 \rfloor x_1 + \left(\lfloor a_2 \rfloor + \frac{f_2 - f_b}{1 - f_b} \right) x_2 - \frac{1}{1 - f_b} y \leq \lfloor b \rfloor$$

is valid for F_{MIR}.

Simplex Algorithm

Simplex finds $\hat{x} \in P \times \mathbb{R}^{N-n}_{\geq 0}$ and creates the optimal simplex tableau:

$i-$th row in the simplex tableau

$$x_{B_i} + \sum_{j \in NB} \bar{a}_{ij} x_j = \bar{b}_i$$

- $x_1, \ldots, x_{n_1}$: Integral decision variables
- $x_{n_1+1}, \ldots, x_n$: Real decision variables
- $x_{n+1}, \ldots, x_N$: (Real) slack variables

- $B = \{B_1, \ldots, B_m\}$: Basic variables
- $NB = \{1, \ldots, N\} \setminus B$: Nonbasic variables ($\hat{x}_j = 0$ for $j \in NB$)

Gomory Mixed Integer Cut

Let $N_1 = NB \cap \{1, \ldots, n_1\}$, $N_2 = NB \cap \{n_1 + 1, \ldots, N\}$. Consider the $i-$th row in the optimal simplex tableau

$$x_{B_i} + \sum_{j \in N_1} \bar{a}_{ij} x_j + \sum_{j \in N_2} \bar{a}_{ij} x_j = \bar{b}_i$$

and assume $B_i \leq n_1$ but $\hat{x}_{B_i} = \bar{b}_i \notin \mathbb{Z}$. Then the <u>Gomory Mixed Integer Cut</u>

$$\boxed{x_{B_i} + \sum_{\substack{j \in N_1 \\ f_{ij} \leq f_i}} \lfloor \bar{a}_{ij} \rfloor x_j + \sum_{\substack{j \in N_1 \\ f_{ij} > f_i}} \left(\lfloor \bar{a}_{ij} \rfloor + \frac{f_{ij} - f_i}{1 - f_i} \right) x_j + \sum_{\substack{j \in N_2 \\ \bar{a}_{ij} < 0}} \left(\frac{\bar{a}_{ij}}{1 - f_i} \right) x_j \leq \lfloor \bar{b}_i \rfloor}$$

is a valid inequality for F that is not satisfied by $\hat{x}$.

Project Demonstration

- Simplex Solver
- Mixed Integer Gomory Cut
- 2D Visualisation

Cutting Planes Selection

- Only adding an arbitrary, single cutting plane is very inefficient if the problem dimension is large.

Cutting Planes Selection

- Only adding an arbitrary, single cutting plane is very inefficient if the problem dimension is large.
  - Evaluate the efficiency of a cutting plane based on some heuristics (for example euclidean distance to $\hat{x}$).

Cutting Planes Selection

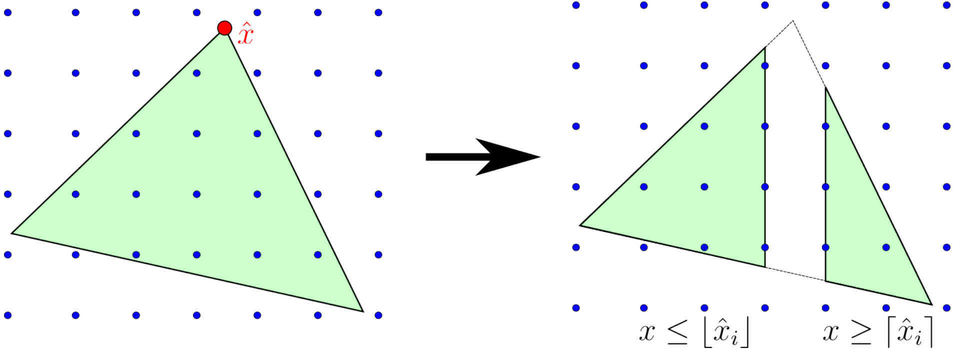
- Only adding an arbitrary, single cutting plane is very inefficient if the problem dimension is large.
  - Evaluate the efficiency of a cutting plane based on some heuristics (for example euclidean distance to $\hat{x}$).
  - Add multiple cutting planes in each iteration.

Cutting Planes Selection

- Only adding an arbitrary, single cutting plane is very inefficient if the problem dimension is large.
  - Evaluate the efficiency of a cutting plane based on some heuristics (for example euclidean distance to $\hat{x}$).
  - Add multiple cutting planes in each iteration.
- For binary programs: Knapsack Covers or GUB (generalized upper bound) covers.

The Branch in Branch & Bound

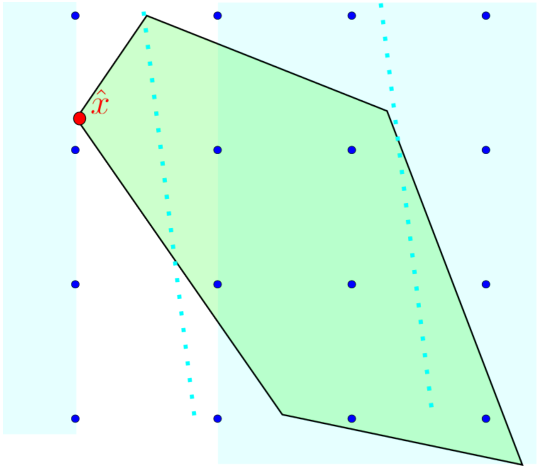
Divide & Conquer: Cut off the non-integer neighborhood of $\hat{x}_i \notin \mathbb{Z}$ and obtain two new relaxation problems, one with $x_i \leq \lfloor \hat{x}_i \rfloor$ and one with $x_i \geq \lceil \hat{x}_i \rceil$. The resulting data structure is a binary tree of problems.



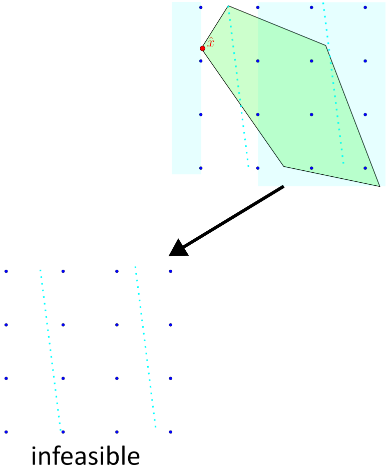
The Bound in Branch & Bound

- Initially, we only know that $-\infty < c^\top x^* < \infty$ (not very helpful).
- Improve bounds:
  - Lower Bound: For any LP-Relaxation, we have $c^\top \hat{x} \leq c^\top x^*$
  - Upper Bound: By definition, $c^\top x^* \leq c^\top x$ for any feasible $x \in F$
- For an optimal solution $\hat{x}$ of a subproblem:
  - if $c^\top \hat{x} \geq$ upper bound, prune tree (stop branching).
  - if $\hat{x}$ is feasible, update upper bound and prune.
  - if $\hat{x}$ is infeasible, update lower bound and branch.
  - stop if tree is completely pruned or upper bound $-$ lower bound $< \epsilon$.

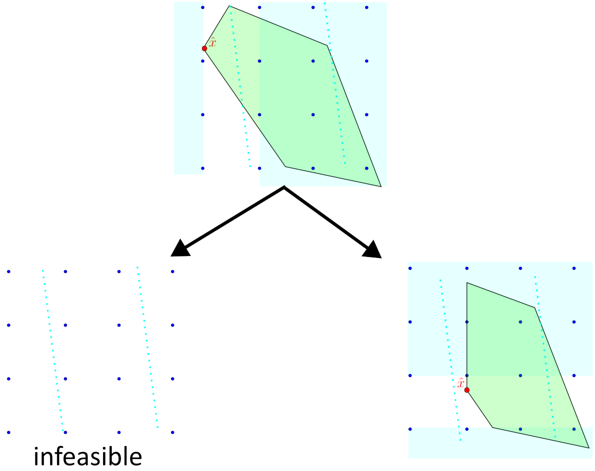
Example



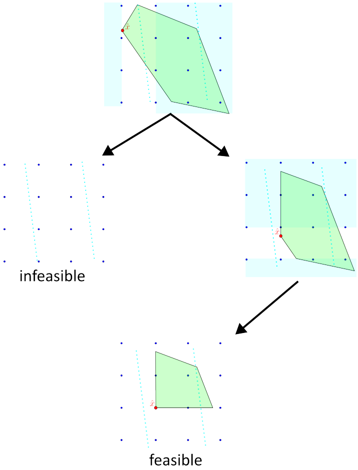
Example



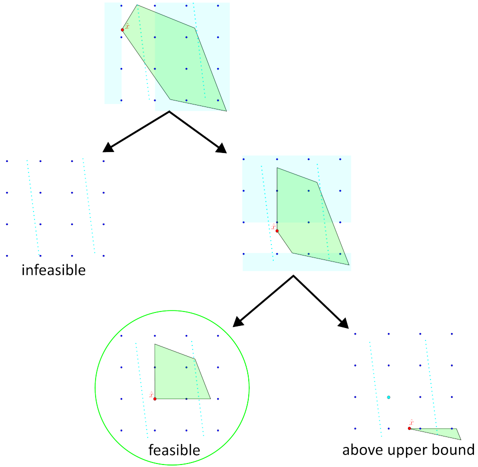
Example



Example



Example



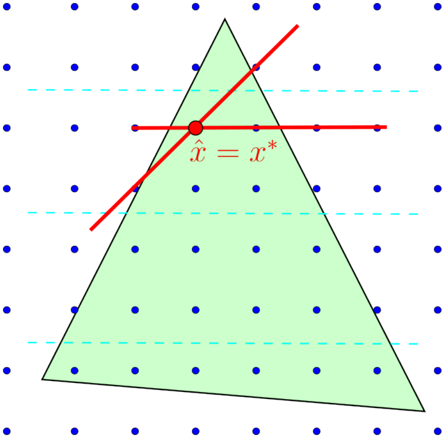
Subproblem and Branching Variable Selection Strategies

- Subproblem selection:
  - Depth-First: Descend quickly to obtain a good upper bound (obtain a feasible solution fast).
  - Best-First: Pick the active node with the current lower bound (obtain a good lower bound).
- Branching variable selection:
  - Most fractional: $i = \arg\max_{1 \leq i \leq n_1} \min(f_i, 1 - f_i)$ (f_i close to $\frac{1}{2}$).
  - Multiple variables at once.

Cutting Planes + Branch & Bound = Branch & Cut

- Cutting Planes or Branch & Bound on their own are inefficient in practice.
- Combine the two to Branch & Cut. This works like Branch & Bound but with additionally adding cuts before branching.
- Used in practice.

Questions?



References

Bixby, E. Robert, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling (2000). "MIP: Theory and Practice — Closing the Gap". In: System Modelling and Optimization. Ed. by M. J. D. Powell and S. Scholtes. Boston, MA: Springer US, pp. 19–49. ISBN: 978-0-387-35514-6.

Goycoolea, Marcos (n.d.). "Cutting Planes for Large Mixed Integer Programming Models". In: ().

"Integer Programming: 2nd Edition" (2020). In: Integer Programming. John Wiley and Sons, Ltd, pp. 1–34. ISBN: 9781119606475. DOI: https://doi.org/10.1002/9781119606475.oth1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119606475.oth1. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119606475.oth1.

Papadimitriou, Christos and Kenneth Steiglitz (Jan. 1982). Combinatorial Optimization: Algorithms and Complexity. Vol. 32. ISBN: 0-13-152462-3. DOI: 10.1109/TASSP.1984.1164450.