Computational Mixed-Integer Programming

Ambros Gleixner and the SCIP team

Zuse Institute Berlin · gleixner@zib.de
SCIP Optimization Suite · http://scip.zib.de

Theory and Practice of Satisfiability Solving

BIRS-CMO · Oaxaca · August 28, 2018

SPONSORED BY THE

Federal Ministry of Education and Research

Zuse Institute Berlin – Fast Algorithms, Fast Computers

A research institute and computing center of the State of Berlin with research units:

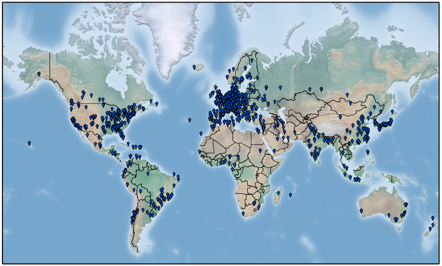
- Numerical Analysis and Modeling
- Visualization and Data Analysis
- Optimization:
  Energy – Transportation – Health – Mathematical Optimization Methods
- Scientific Information Systems
- Computer Science and High Performance Computing

SCIP: Solving Constraint Integer Programs

An open branch-cut-and-price framework with techniques from MIP, CP, SAT, and GO.

35+ active developers

- 10+ running Bachelor & Master projects
- 14+ running PhD projects
- 11 postdocs and professors



5 active development centers

- ZIB: SCIP, SoPlex, UG, ZIMPL
- TU Darmstadt: SCIP and SCIP-SDP
- FAU Erlangen-Nürnberg: SCIP
- RWTH Aachen & Uni. Lancaster: GCG

Many international contributors and users

- more than 14 000 downloads per year from 100+ countries

Careers

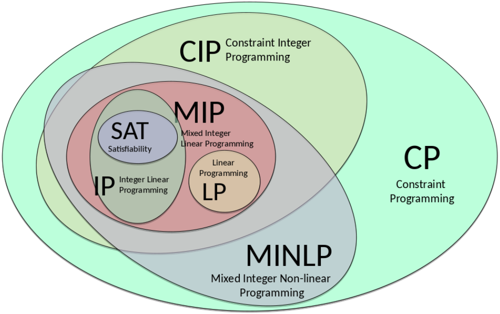
- 7 former developers are now building commercial optimization software at CPLEX, FICO Xpress, Gurobi, MOSEK, and GAMS
- 10 awards for Masters and PhD theses: MOS, EURO, GOR, DMV

Mixed-Integer Programming

A generalization of SAT:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{Z}^I_{\geq 0} \times \mathbb{R}^C_{\geq 0} \end{aligned}$$

1. general linear constraints
2. general integer variables
3. continuous variables
4. objective function



Appealing for similar and different reasons than SAT:

- black-box solvers exist ⇝ separates modeling and algorithm design
- mostly open-box solvers ⇝ allows for problem-specific improvements
- global optimality guarantees ⇝ allows to stop early at near-optimal solutions
- ...

Outline

Essentials

- Linear programming relaxation
- LP-based branch-and-bound
- Cutting planes
- Simplex hot starts

Supplementary techniques

- Presolving & propagation
- Conflict analysis
- Branching heuristics
- Node selection
- Primal heuristics
- Symmetry handling

Numerics & exact certificates

- Numerics
- Verifying MIP results

Conclusion

Outline

Relaxations and bounds

A common approach for hard nonconvex optimization problems like MIP: compute bounds on the optimal value

$$\begin{aligned} z^* = \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{Z}^I_{\geq 0} \times \mathbb{R}^C_{\geq 0} \end{aligned}$$

Relaxations and bounds

A common approach for hard nonconvex optimization problems like MIP: compute bounds on the optimal value

$$\begin{aligned} z^* = \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{Z}^I_{\geq 0} \times \mathbb{R}^C_{\geq 0} \end{aligned}$$

1. Lower bound $L \leq z^*$: relaxation
   - in MIP: **LP relaxation**, $\mathbb{Z}^I \rightsquigarrow \mathbb{R}^I$
   - convex and “fast” to solve $\rightsquigarrow x^{\text{LP}}$

Relaxations and bounds

A common approach for hard nonconvex optimization problems like MIP: compute bounds on the optimal value

$$\begin{aligned} z^* = \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{Z}^I_{\geq 0} \times \mathbb{R}^C_{\geq 0} \end{aligned}$$

1. Lower bound $L \leq z^*$: relaxation
   - in MIP: **LP relaxation**, $\mathbb{Z}^I \rightsquigarrow \mathbb{R}^I$
   - convex and "fast" to solve $\rightsquigarrow x^{\text{LP}}$
2. Upper bound $U \geq z^*$: feasible solutions
   - if LP relaxation is "accidentally" feasible $\rightsquigarrow$ optimal solution
   - later: primal heuristics

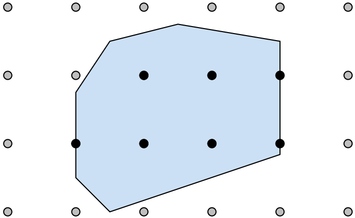
Outline

LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree

Solution space

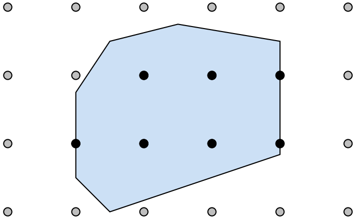


LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree

Solution space

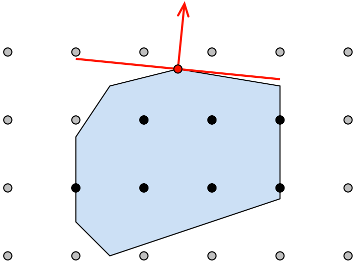


LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree

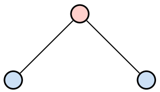
Solution space



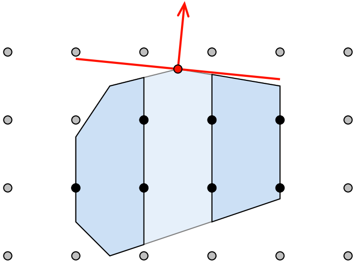
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



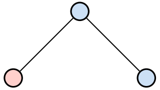
Solution space



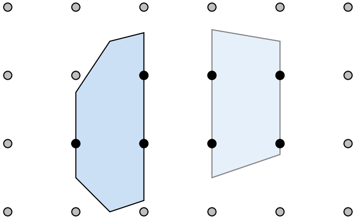
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



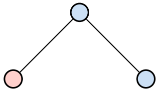
Solution space



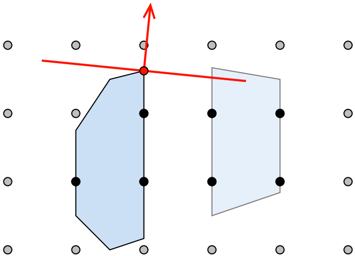
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



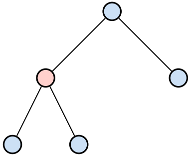
Solution space



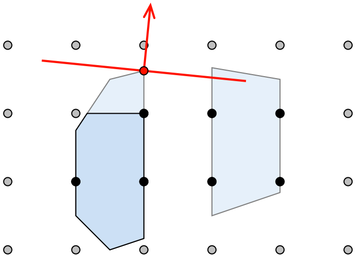
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



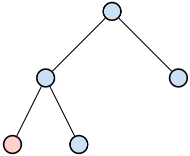
Solution space



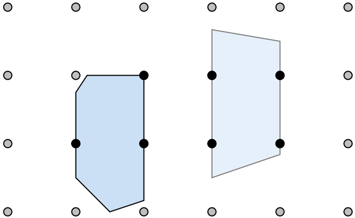
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



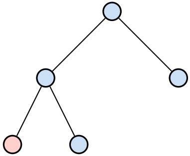
Solution space



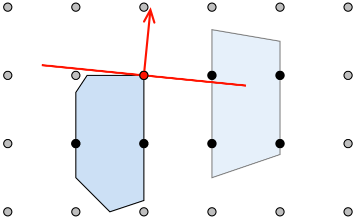
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



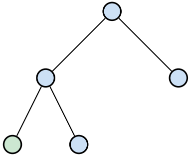
Solution space



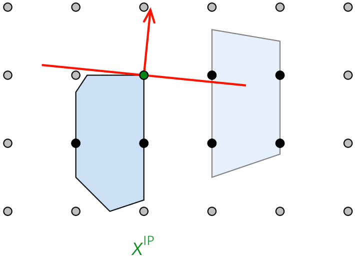
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



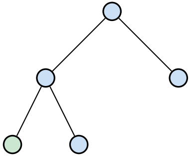
Solution space



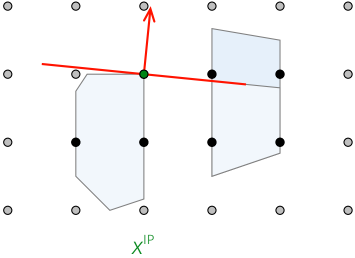
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



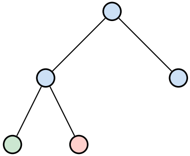
Solution space



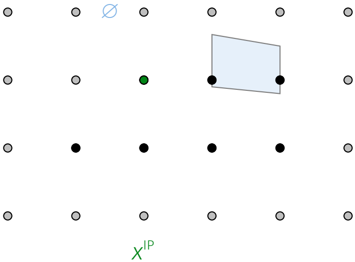
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



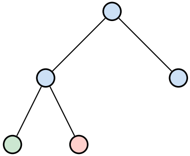
Solution space



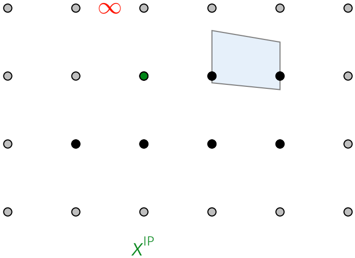
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



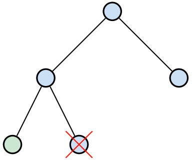
Solution space



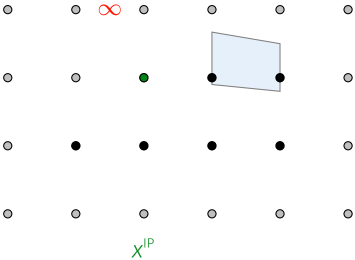
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



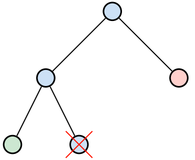
Solution space



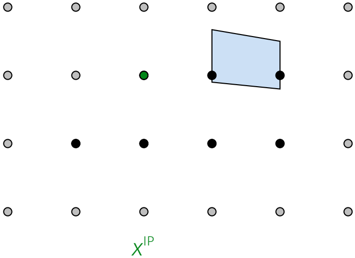
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



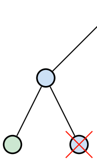
Solution space



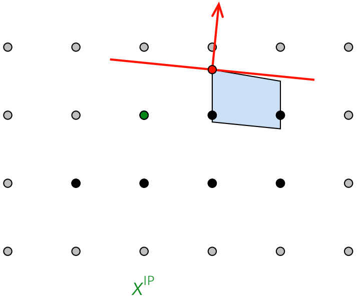
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



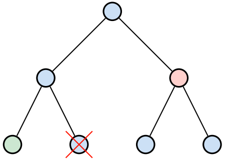
Solution space



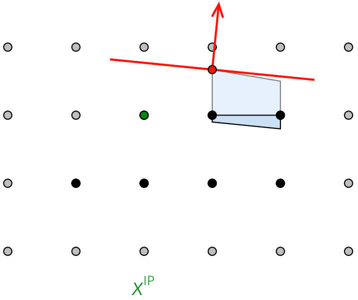
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



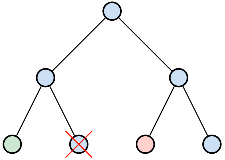
Solution space



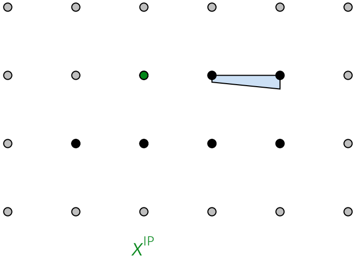
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



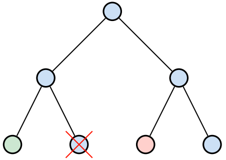
Solution space



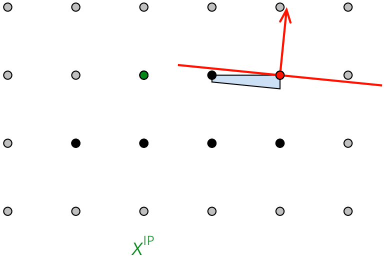
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



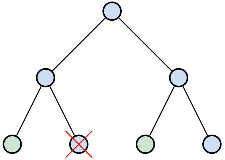
Solution space



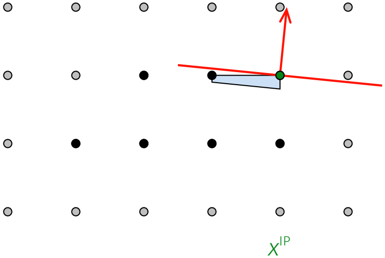
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



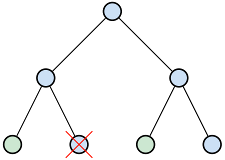
Solution space



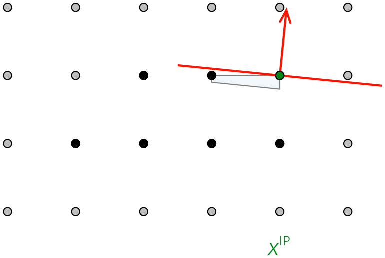
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



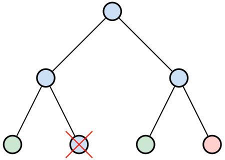
Solution space



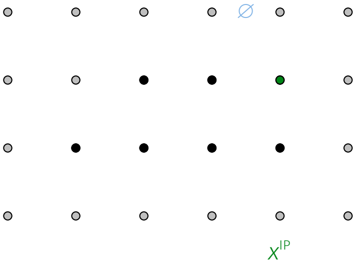
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



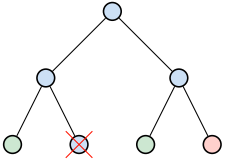
Solution space



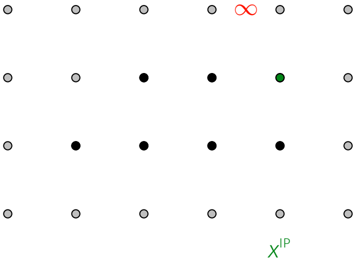
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



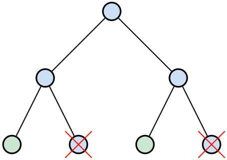
Solution space



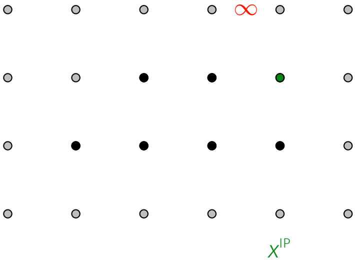
LP-based branch-and-bound

Systematic reduction of $U - L$ by divide-and-conquer [LD60, Dak65]

Branch-and-bound tree



Solution space



LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: “best bound”

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: "best bound"
3. termination:
    - x^{LP} integer $\Rightarrow$ improve "incumbent" U

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: "best bound"
3. termination:
    - x^{LP} integer $\Rightarrow$ improve "incumbent" U
    - node LP infeasible $\Rightarrow$ prune

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: “best bound”
3. termination:
   - x^{LP} integer $\Rightarrow$ improve “incumbent” U
   - node LP infeasible $\Rightarrow$ prune
   - node LP bound $> U \Rightarrow$ prune

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: "best bound"
3. termination:
    - x^{LP} integer $\Rightarrow$ improve "incumbent" U
    - node LP infeasible $\Rightarrow$ prune
    - node LP bound $> U \Rightarrow$ prune
    - stop when $U \leq L$

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: “best bound”
3. termination:
   - x^{LP} integer $\Rightarrow$ improve “incumbent” U
   - node LP infeasible $\Rightarrow$ prune
   - node LP bound $> U \Rightarrow$ prune
   - stop when $U \leq L$
4. proven optimality gap $g = U - L$ before termination

LP-based branch-and-bound

Details

1. the union of leaf nodes contains all improving solutions
2. L = smallest LP bound over all leaf nodes: "best bound"
3. termination:
    - x^{LP} integer $\Rightarrow$ improve "incumbent" U
    - node LP infeasible $\Rightarrow$ prune
    - node LP bound $> U \Rightarrow$ prune
    - stop when $U \leq L$
4. proven optimality gap $g = U - L$ before termination
5. branching on general constraints can yield smaller trees, but
    - increases and slows down the LP, and
    - the high degree of freedom makes designing branching heuristics challenging,

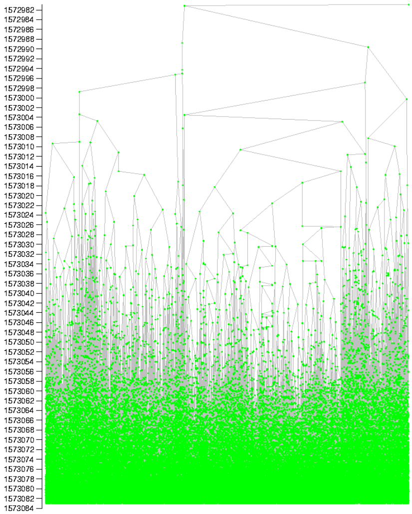
   so by default MIP solvers rely on variable-based branching, see [GMB+15] and references therein.

Example: the 15,112 cities TSP

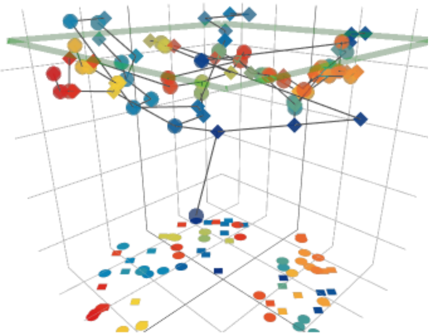
Schematic tree for a 15,112 cities traveling salesman problem

(World record 2001 by Applegate, Bixby, Chvátal, Cook)

`http://www.math.uwaterloo.ca/tsp/d15sol/`



Example: MIPLIB instance lseu



Spatial visualization of the branch-and-cut tree using multi-dimensional scaling, due to Matthias Miltenberger

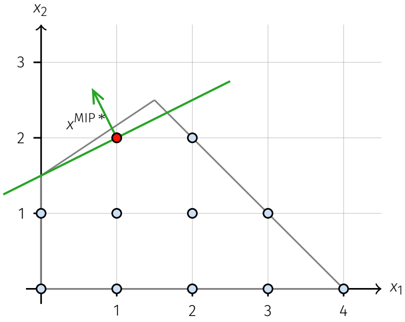
`http://www.zib.de/miltenberger/plotly`

Outline

Cutting planes

$$\mathcal{X}_{\text{MIP}} := \{x \in \mathbb{Z}^I \times \mathbb{R}^C : Ax \leq b\}$$

$$\mathcal{X}_{\text{LP}} := \{x \in \mathbb{R}^I \times \mathbb{R}^C : Ax \leq b\}$$



$$\min\{c^T x : x \in \mathcal{X}_{\text{MIP}}\}$$

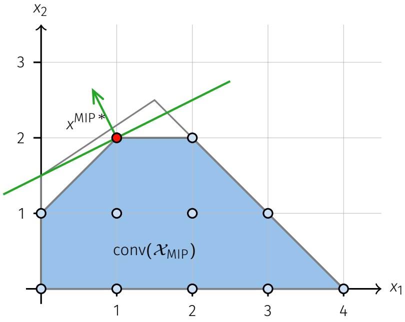
Cutting planes

Observation

- $\text{conv}(\mathcal{X}_{\text{MIP}})$ is a polyhedron
- IP could be formulated as LP

Problems with $\text{conv}(\mathcal{X}_{\text{MIP}})$:

- linear description not known
- large no. of constraints



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

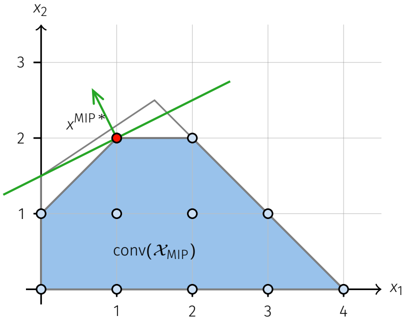
Cutting planes

Observation

- $\text{conv}(\mathcal{X}_{\text{MIP}})$ is a polyhedron
- IP could be formulated as LP

Problems with $\text{conv}(\mathcal{X}_{\text{MIP}})$:

- linear description not known
- large no. of constraints



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

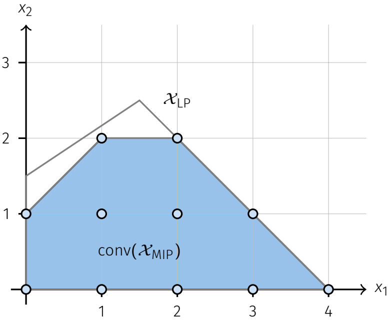
$$\mathcal{X}_{\text{LP}} \supseteq \mathcal{X} \supseteq \text{conv}(\mathcal{X}_{\text{MIP}})$$

$$\min\{c^\top x : x \in \mathcal{X}_{\text{LP}}\} \leq \min\{c^\top x : x \in \mathcal{X}\} = \min\{c^\top x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{rr} \min & c^\top x \\ s.t. & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for conv($\mathcal{X}_{\text{MIP}}$) but
   - violated by x^*.
5. Goto 2.

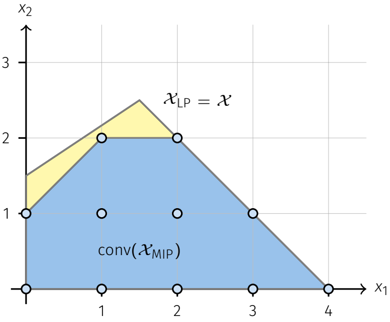


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\mathrm{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ s.t. & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\mathrm{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\mathrm{conv}(\mathcal{X}_{\mathrm{MIP}})$ but
   - violated by x^*.
5. Goto 2.

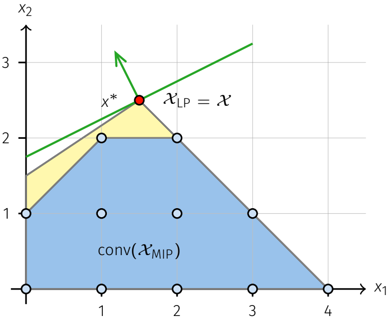


$\min\{c^T x : x \in \mathrm{conv}(\mathcal{X}_{\mathrm{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.



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

General cutting plane method

Algorithm

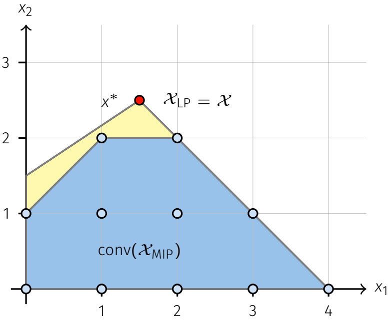
1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$

2. Solve
$$\begin{array}{ll} \min & c^\top x \\ s.t. & x \in \mathcal{X} \end{array}$$

3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop

4. Add inequality to $\mathcal{X}$ that is ...
   - valid for conv($\mathcal{X}_{\text{MIP}}$) but
   - violated by x^*.

5. Goto 2.

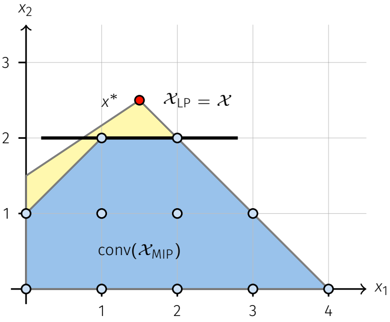


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ s.t. & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

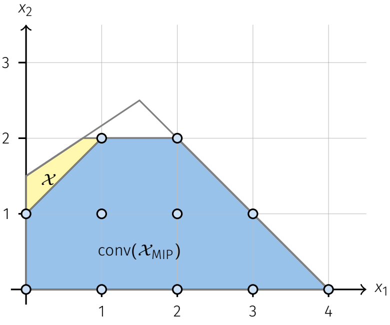


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

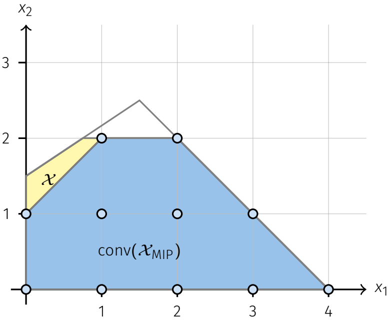


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{rr} \min & c^\top x \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

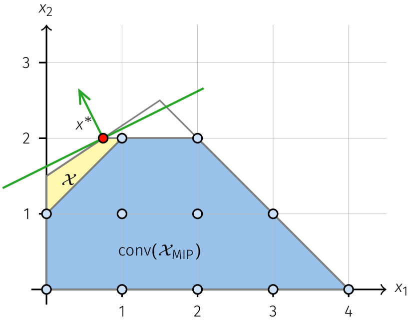


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\mathrm{LP}}$
2. Solve
$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$
3. If $x^* \in \mathcal{X}_{\mathrm{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\mathrm{conv}(\mathcal{X}_{\mathrm{MIP}})$ but
   - violated by x^*.
5. Goto 2.

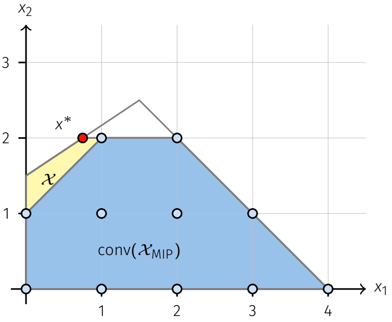


$\min\{c^T x : x \in \mathrm{conv}(\mathcal{X}_{\mathrm{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ s.t. & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

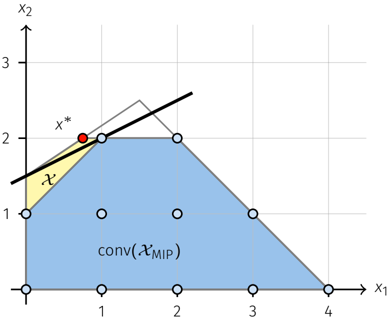


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

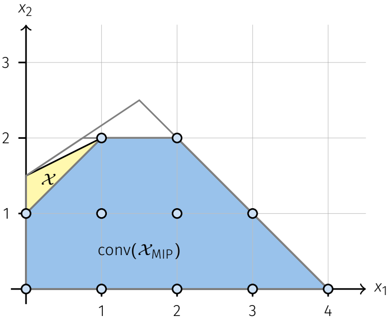


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{rr} \min & c^\top x \\ s.t. & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

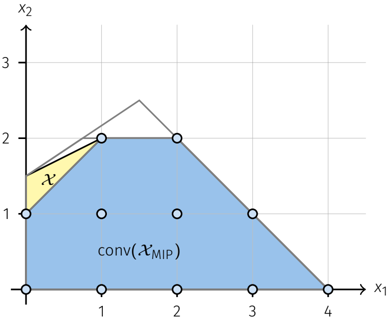


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

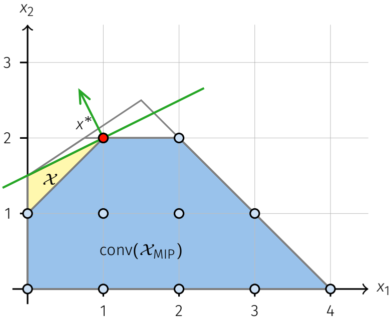


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
    - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
    - violated by x^*.
5. Goto 2.

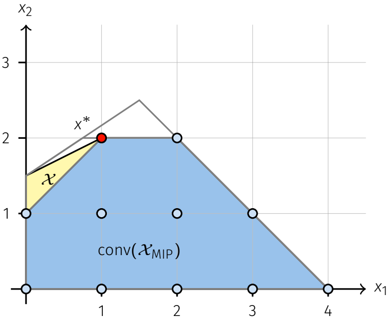


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for $\text{conv}(\mathcal{X}_{\text{MIP}})$ but
   - violated by x^*.
5. Goto 2.

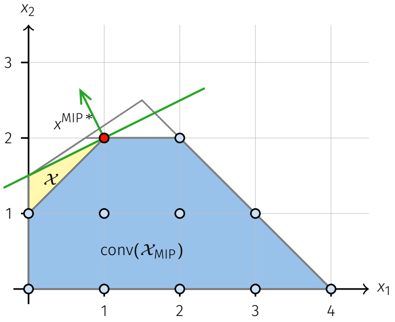


$\min\{c^T x : x \in \text{conv}(\mathcal{X}_{\text{MIP}})\}$

General cutting plane method

Algorithm

1. $\mathcal{X} \leftarrow \mathcal{X}_{\text{LP}}$
2. Solve
$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$
3. If $x^* \in \mathcal{X}_{\text{MIP}}$: Stop
4. Add inequality to $\mathcal{X}$ that is ...
   - valid for conv($\mathcal{X}_{\text{MIP}}$) but
   - violated by x^*.
5. Goto 2.



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

- Gomory cuts yield a finitely convergent, pure cutting plane algorithm [Gom58].

Example: knapsack cover cuts

Start from single-row relaxation

$$X := \{\, x \in \{0,1\}^n \;:\; \sum_{j \in N} a_j x_j \leq b \,\}$$

for $b \in \mathbb{Z}_{>0}$, $a_j \in \mathbb{Z}_{>0}\;\; \forall j \in N$.

Minimal cover $C \subseteq N \Leftrightarrow \sum_{j \in C} a_j > b$ and $\sum_{j \in C \setminus \{i\}} a_j \leq b \quad \forall i \in C$.

$$\implies \sum_{j \in C} x_j \leq |C| - 1$$

Example: knapsack cover cuts

Start from single-row relaxation

$$X := \{\, x \in \{0,1\}^n \;:\; \sum_{j\in N} a_j x_j \le b \,\}$$

for $b \in \mathbb{Z}_{>0}$, $a_j \in \mathbb{Z}_{>0} \;\; \forall j \in N$.

Minimal cover $C \subseteq N$ $\Leftrightarrow \sum_{j\in C} a_j > b$ and $\sum_{j\in C\setminus\{i\}} a_j \le b \quad \forall i \in C$.

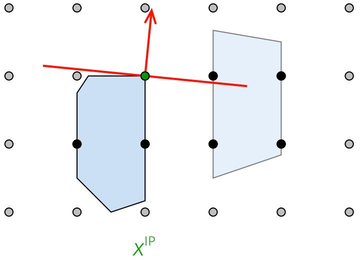
$$\Longrightarrow \sum_{j\in C} x_j \le |C| - 1$$

Separation problem

$$\min \Big\{ \sum_{j\in N} (1 - x_j^*) y_j : \sum_{j\in N} a_j y_j \ge b + 1, y_j \in \{0,1\} \Big\}$$

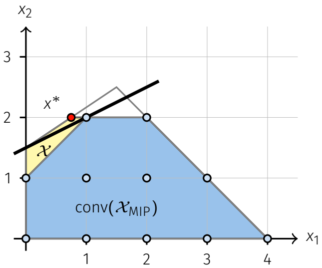
yields the most violated cut for given x^* (typically solved heuristically).

Branch-and-cut



branch-and-bound

+



cutting planes

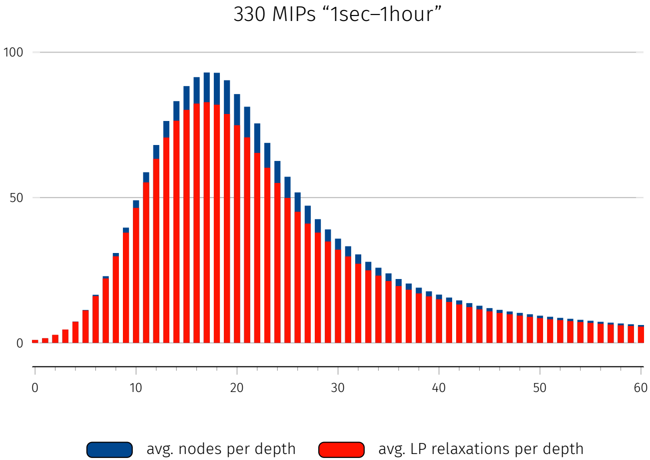
- limited cutting plane generation at root and nodes of a branch-and-bound tree
- cut selection and aging crucial

Outline

Dual simplex iterations during tree search

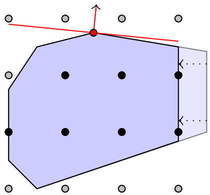
383.7/3.3 ≈ 116x speedup

depth	avg. iters
root	**383.7**
1	17.0
2	14.9
3	12.2
4	10.3
5	9.0
6	8.2
⋮	⋮
13	4.2
14	4.1
15	3.8
16	3.4
17	3.3
18	3.1
19	2.8
20	2.7
21	2.5
22	2.4
⋮	⋮



Outline

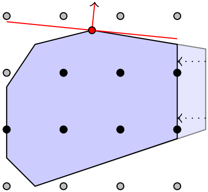
Presolving and propagation



Task

- reduce size of model by removing irrelevant information
- strengthen LP relaxation by exploiting integrality information
- make the LP relaxation numerically more stable
- extract useful information

Presolving and propagation



Task

- reduce size of model by removing irrelevant information
- strengthen LP relaxation by exploiting integrality information
- make the LP relaxation numerically more stable
- extract useful information

Primal Reductions:
- based on feasibility reasoning
- no feasible solution is cut off

Weak/strong dual reductions:
- consider objective function
- all/at least one optimal solution remains

Trivial presolving

Fast and useful:

- remove empty rows, columns
  - e.g., $0^T x \leq b_i,\ b_i < 0 \Rightarrow$ infeasible
- tighten fractional bounds of integer variables
- substitute fixed variables
- replace singleton rows
  - e.g., $a_{ij}x_j \leq b_i,\ a_{ij} < 0 \Rightarrow x_j \geq \frac{b_i}{a_{ij}} \Rightarrow$ new lower bound on x_j
- normalize constraints
  - e.g., if all coefficients are integral, divide by greatest common divisor and round rhs
- detect constraint types (knapsack, setppc, …)
- …

Linear presolving

Let a linear constraint $a^T x \le b$, and bounds $\ell \le x \le u$ be given.

Linear presolving

Let a linear constraint $a^T x \leq b$, and bounds $\ell \leq x \leq u$ be given.

$$\alpha_{\min} \quad := \quad \begin{array}{ll} \min & a^T x \\ s.t. & \ell \leq x \leq u \end{array} \quad = \quad \textstyle\sum_{j,a_j>0} a_j \ell_j + \sum_{j,a_j<0} a_j u_j.$$

and

Linear presolving

Let a linear constraint $a^T x \leq b$, and bounds $\ell \leq x \leq u$ be given.

$$\alpha_{\min} := \begin{array}{rc} \min & a^T x \\ s.t. & \ell \leq x \leq u \end{array} = \sum_{j, a_j > 0} a_j \ell_j + \sum_{j, a_j < 0} a_j u_j.$$

and

$$\alpha_{\max} := \begin{array}{rc} \max & a^T x \\ s.t. & \ell \leq x \leq u \end{array} = \sum_{j, a_j < 0} a_j \ell_j + \sum_{j, a_j > 0} a_j u_j.$$

are the minimal and maximal activity of the linear constraint.

Linear presolving

Let a linear constraint $a^T x \leq b$, and bounds $\ell \leq x \leq u$ be given.

$$\alpha_{\min} \quad := \quad \begin{array}{rc} \min & a^T x \\ s.t. & \ell \leq x \leq u \end{array} \quad = \quad \textstyle\sum_{j,a_j>0} a_j \ell_j + \sum_{j,a_j<0} a_j u_j.$$

and

$$\alpha_{\max} \quad := \quad \begin{array}{rc} \max & a^T x \\ s.t. & \ell \leq x \leq u \end{array} \quad = \quad \textstyle\sum_{j,a_j<0} a_j \ell_j + \sum_{j,a_j>0} a_j u_j.$$

are the minimal and maximal activity of the linear constraint.

First observation

- $\alpha_{\min} > b \Rightarrow$ problem infeasible
- $\alpha_{\max} \leq b \Rightarrow$ constraint redundant

Bound strengthening

Let $a_k > 0$. For all feasible solutions x, it holds that:

$$a^T x - a_k x_k + a_k x_k \leq b \Leftrightarrow x_k \leq \frac{b-(a^T x - a_k x_k)}{a_k} \Rightarrow x_k \leq \frac{b-\alpha_{\min}+a_k \ell_k}{a_k}$$

Bound strengthening

Let $a_k > 0$. For all feasible solutions x, it holds that:

$$a^T x - a_k x_k + a_k x_k \leq b \Leftrightarrow x_k \leq \frac{b-(a^T x - a_k x_k)}{a_k} \Rightarrow x_k \leq \frac{b-\alpha_{\min}+a_k \ell_k}{a_k}$$

$$\Rightarrow x_k \leq \min\left\{u_k, \frac{b-\alpha_{\min}+a_k \ell_k}{a_k}\right\}$$

Variants:

- $k \in I \Rightarrow x_k \leq \lfloor \frac{b-\alpha_{\min}+a_k \ell_k}{a_k} \rfloor$
- $a_k < 0 \Rightarrow x_k \geq \frac{b-\alpha_{\max}+a_k u_k}{a_k}$

Global information: the conflict graph [ANS00]

For the set of binary variables $\mathcal{B}$ in a MIP M, the conflict or clique graph is the undirected $G = (V, E)$ with nodes

$$V := \mathcal{B} \times \{0, 1\} = \{j_\kappa, j \in \mathcal{B}, \kappa \in \{0, 1\}\}.$$

and edges

$$\begin{aligned} E := \{\{v, w\} :& \kappa_v x_v + \kappa_w x_w + \\ & (1 - \kappa_v)(1 - x_v) + (1 - \kappa_w)(1 - x_w) \leq 1 \text{ valid for } M\} \end{aligned}$$

Global information: the conflict graph [ANS00]

For the set of binary variables $\mathcal{B}$ in a MIP M, the conflict or clique graph is the undirected $G = (V, E)$ with nodes

$$V := \mathcal{B} \times \{0, 1\} = \{j_\kappa, j \in \mathcal{B}, \kappa \in \{0, 1\}\}.$$

and edges

$$E := \{\{v, w\} : \kappa_v x_v + \kappa_w x_w + (1 - \kappa_v)(1 - x_v) + (1 - \kappa_w)(1 - x_w) \leq 1 \text{ valid for } M\}$$

Example

Let $V = \{1, 2, 3\} \times \{0, 1\}$, and let E consist of the following edges:

1. $\{1_1, 2_1\} \Rightarrow x_1 + x_2 \leq 1$
2. $\{2_1, 3_0\} \Rightarrow x_2 + (1 - x_3) \leq 1$
3. $\{3_1, 1_1\} \Rightarrow x_3 + x_1 \leq 1$

The conflict graph is first populated during presolving and used for separation and propagation during the solving process.

More presolving

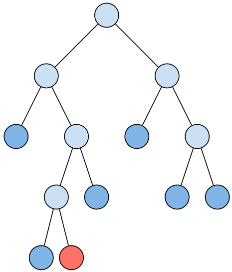
- probing: tentatively fix binary variables and propagate
- dominance test: pairwise comparison of rows/columns
- aggregation of equations with only two variables
  $a_k x_k + a_j x_j = b \Rightarrow x_k = \frac{b}{ai} - \frac{a_j}{ak} x_j$
- dual fixing: If $a_{ik} \geq 0$ for all i and $c_k \geq 0$, then x_k can be fixed to its lower bound
- dual aggregation: If $c_k \geq 0$ and there is exactly one i for which $a_{ik} < 0$, we can aggregate $x_k = \frac{b_i}{a_j} - \frac{1}{a_k} \sum a_j x_j$.
- dual bound reduction: Strengthen bounds of variables to the tightest value for which all its constraints are redundant
- clique detection (e.g., for knapsack constraints)
- variable lifting
- ...

Outline

Introduction: Conflict Analysis

Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

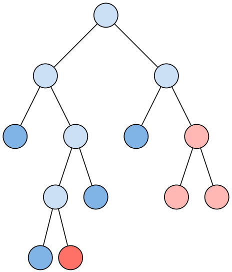
- extract a shorter reason



Introduction: Conflict Analysis

Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

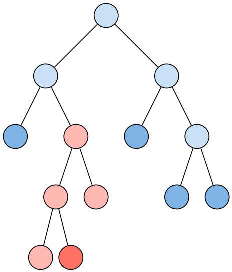
- extract a shorter reason
- that prunes other parts of the tree



Introduction: Conflict Analysis

Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

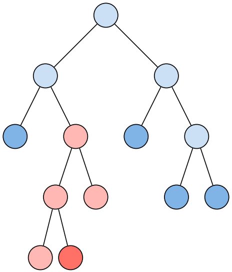
- extract a shorter reason
- that prunes other parts of the tree
- also in backtracking



Introduction: Conflict Analysis

Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

- extract a shorter reason
- that prunes other parts of the tree
- also in backtracking



Example: contradicting bound changes after propagation

$$x_1 + x_2 + 2x_3 \leq 2 \qquad (1)$$

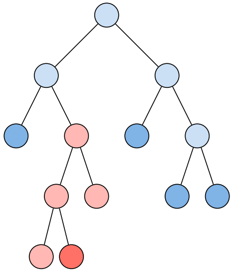
$$x_1 + x_2 - 2x_3 \leq 0 \qquad (2)$$

$$x_1, x_2, x_3 \in \{0, 1\}$$

Introduction: Conflict Analysis

Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

- extract a shorter reason
- that prunes other parts of the tree
- also in backtracking

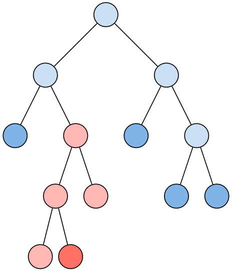


Example: contradicting bound changes after propagation

$$
\begin{aligned}
x_1 + x_2 + 2x_3 &\le 2 \qquad (1) \\
x_1 + x_2 - 2x_3 &\le 0 \qquad (2) \\
x_1, x_2, x_3 &\in \{0, 1\}
\end{aligned}
$$

$$(x_2 \ge 1) \overset{(1)}{\Longrightarrow} (x_3 \le 0)$$

Introduction: Conflict Analysis



Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

- extract a shorter reason
- that prunes other parts of the tree
- also in backtracking

Example: contradicting bound changes after propagation

$$x_1 + x_2 + 2x_3 \le 2 \qquad (1)$$
$$x_1 + x_2 - 2x_3 \le 0 \qquad (2)$$
$$x_1, x_2, x_3 \in \{0, 1\}$$

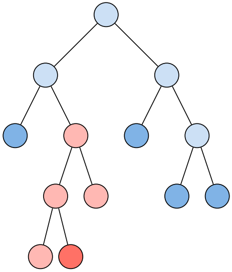
$(x_2 \ge 1) \overset{(1)}{\Longrightarrow} (x_3 \le 0)$

$\Longrightarrow$ (2) is violated

Introduction: Conflict Analysis

Goal: When branch-and-bound reaches an infeasible subproblem, analyze the infeasibility to

- extract a shorter reason
- that prunes other parts of the tree
- also in backtracking



Example: contradicting bound changes after propagation

$$x_1 + x_2 + 2x_3 \leq 2 \qquad (1)$$

$$x_1 + x_2 - 2x_3 \leq 0 \qquad (2)$$

$$x_1, x_2, x_3 \in \{0, 1\}$$

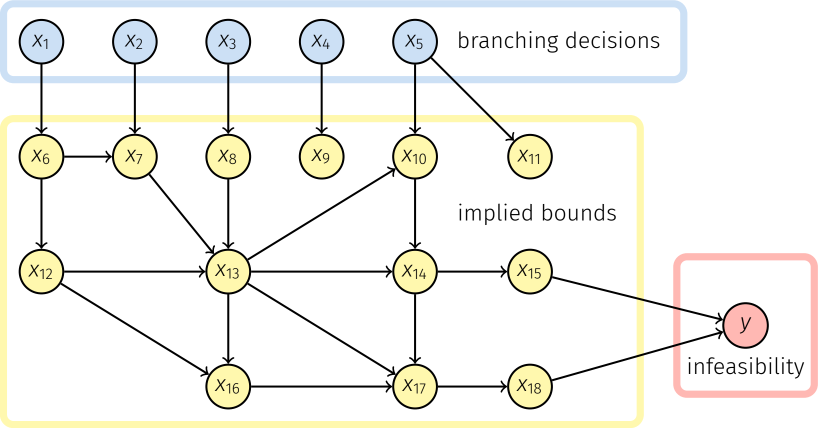
$(x_2 \geq 1) \overset{(1)}{\Longrightarrow} (x_3 \leq 0)$

$\Longrightarrow$ (2) is violated

$\rightsquigarrow$ $x_2 \geq 1$ is a sufficient reason

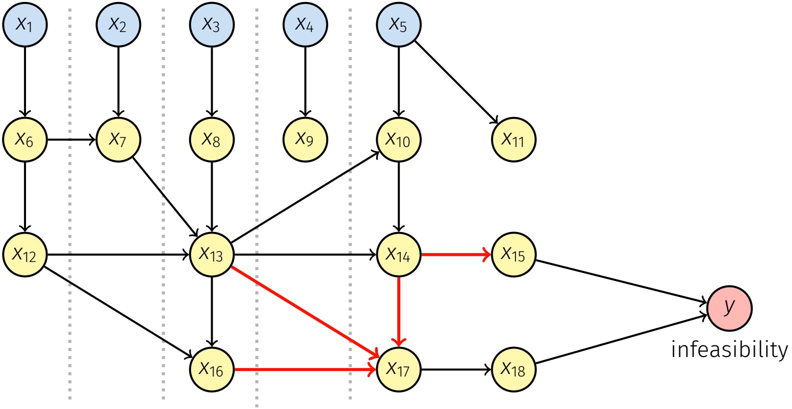
Conflict Graph Analysis [MSS99]

- Consider implications that led to the local bounds



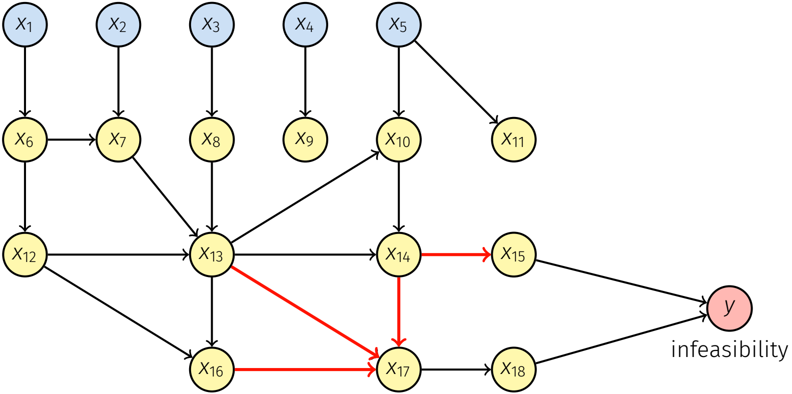
Conflict Graph Analysis [MSS99]

- Consider implications that led to the local bounds
- Each cut that separates branching nodes from y yields a conflict (FUIP, ...)



Conflict Graph Analysis [MSS99]

- Consider implications that led to the local bounds
- Each cut that separates branching nodes from y yields a conflict (FUIP, ...)
- Special: graph is not maintained, but constructed when needed (in SCIP)



Explaining LP infeasibility [SS06, Ach07]

- Assume a subproblem with bounds $\ell \leq \ell' \leq u' \leq u$

$$\min\{c^T x \mid Ax \geq b,\ \ell' \leq x \leq u',\ x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}\} \tag{3}$$

Explaining LP infeasibility [SS06, Ach07]

- Assume a subproblem with bounds $\ell \leq \ell' \leq u' \leq u$

$$\min\{c^T x \mid Ax \geq b,\ \ell' \leq x \leq u',\ x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}\} \tag{3}$$

- LP of (3) infeasible $\Longleftrightarrow$ unbounded direction in the dual

$$\max\{y^T b + r^T\{\ell', u'\} \mid y^T A + r^T = c^T,\ y \in \mathbb{R}^m_+, r \in \mathbb{R}^n\} \tag{4}$$

- i.e. a ray (y, s)

$$\begin{aligned} y^t A + s^t &= 0 \\ y^t b + s^t\{\ell', u'\} &> 0 \end{aligned}$$

Explaining LP infeasibility [SS06, Ach07]

- Assume a subproblem with bounds $\ell \leq \ell' \leq u' \leq u$

$$\min\{c^T x \mid Ax \geq b,\ \ell' \leq x \leq u',\ x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}\} \tag{3}$$

- LP of (3) infeasible $\Longleftrightarrow$ unbounded direction in the dual

$$\max\{y^T b + r^T\{\ell', u'\} \mid y^T A + r^T = c^T,\ y \in \mathbb{R}^m_+, r \in \mathbb{R}^n\} \tag{4}$$

- i.e. a ray (y, s)

$$y^t A + s^t = 0$$
$$y^t b + s^t\{\ell', u'\} > 0$$

$$\Downarrow$$

Option 1. $\{x_i \geq \ell'_i : s_i > 0\} \cup \{x_i \leq u'_i : s_i < 0\}$ (initial conflict)

Option 2. $(y^T A)x \geq y^T b$ (Farkas constraint)

Explaining LP infeasibility [SS06, Ach07]

- Assume a subproblem with bounds $\ell \leq \ell' \leq u' \leq u$

$$\min\{c^T x \mid Ax \geq b,\ \ell' \leq x \leq u',\ x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}\} \tag{3}$$

- LP of (3) infeasible $\Longleftrightarrow$ unbounded direction in the dual

$$\max\{y^T b + r^T\{\ell', u'\} \mid y^T A + r^T = c^T,\ y \in \mathbb{R}^m_+, r \in \mathbb{R}^n\} \tag{4}$$

- i.e. a ray (y, s)

$$y^t A + s^t = 0$$
$$y^t b + s^t\{\ell', u'\} > 0$$

$$\Downarrow$$

Option 1. $\{x_i \geq \ell'_i : s_i > 0\} \cup \{x_i \leq u'_i : s_i < 0\}$ (initial conflict)

Option 2. $(y^T A)x \geq y^T b$ (Farkas constraint)

- Farkas constraint globally valid: propagate during tree search, strengthen via MIR rounding, ... [WBH17]
- analogous extension for bound exceeding LPs

Outline

Basic definitions for variable branching

Notation	Description
P	LP relaxation feasible region $$P := \{x \in \mathbb{R}^n_{\geq 0} : Ax \leq b\}$$
$\mathcal{F}$	Set of fractional variables (fractionality $\neq 0$) $$\mathcal{F} := \{j \in I : x_j^{\mathrm{LP}} \notin \mathbb{Z}\}$$
+,-	Branching directions

Fractionality: distance between LP solution value and branching bound

$$f_j^+ := \lceil x_j^{\mathrm{LP}} \rceil - x_j^{\mathrm{LP}}, \quad f_j^- := x_j^{\mathrm{LP}} - \lfloor x_j^{\mathrm{LP}} \rfloor$$

Most/least infeasible branching

Idea: Select fractional variable with highest fractionality

$$j \in \underset{j' \in \mathcal{F}}{\operatorname{argmax}} \{\min\{f_{j'}^-, f_{j'}^+\}\}$$

or lowest fractionality

$$j \in \underset{j' \in \mathcal{F}}{\operatorname{argmin}} \{\min\{f_{j'}^-, f_{j'}^+\}\}.$$

Most/least infeasible branching

Idea: Select fractional variable with highest fractionality

$$j \in \operatorname*{argmax}_{j' \in \mathcal{F}} \{\min\{f_{j'}^-, f_{j'}^+\}\}$$

or lowest fractionality

$$j \in \operatorname*{argmin}_{j' \in \mathcal{F}} \{\min\{f_{j'}^-, f_{j'}^+\}\}.$$

Problem

LP degeneracy (multiple optimal node LP solutions) on many problems makes fractionality a weak variable attribute.

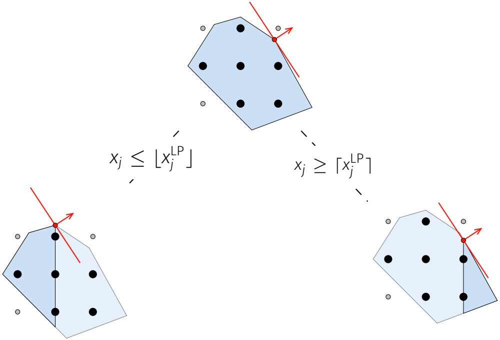
Poor branching performance yielding large trees, sometimes worse than randomized variable selection.

Dual gain

Branching children/descendants:

$$P_j^- := P \cap \{x_j \leq \lfloor x_j^{\mathrm{LP}} \rfloor\}, P_j^+ := P \cap \{x_j \geq \lceil x_j^{\mathrm{LP}} \rceil\}$$

Dual gain: LP objective between a descendant and its parent node P:



$$\Delta c_j^* := \min\{c^T x : x \in P_j^*\}\} - \min\{c^T x : x \in P\} \geq 0, \quad * \in \{-, +\}$$

Scoring function

Selecting fractional candidates based on scores for individual directions

$$s^- := \Delta c_j^-, s^+ := \Delta c_j^+ \forall j \in \mathcal{F}$$

requires scoring function:
$s(s^-, s^+)\colon \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$

Possibilities:

- Weighted sum for $\lambda \in [0, 1]$:

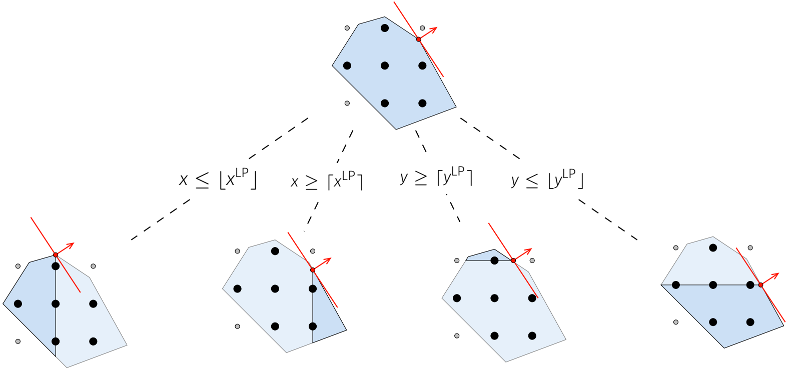
$$s(s^-, s^+) := \lambda \max\{s^-, s^+\} + (1 - \lambda) \min\{s^-, s^+\}$$

- Product for small $\epsilon > 0$:

$$s(s^-, s^+) := \max\{s^-, \epsilon\} \cdot \max\{s^-, \epsilon\}$$

Lookahead: strong branching

1. Perform an explicit look-ahead by solving all possible descendants of the current node.



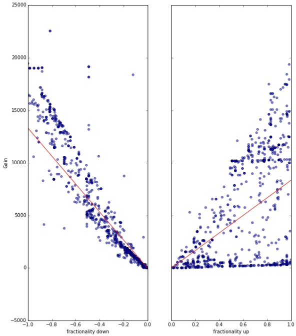
2. Select a fractional variable $j \in \operatorname{argmax}_{j' \in \mathcal{F}} \{ s\{\Delta c_{j'}^{-}, \Delta c_{j'}^{+}\}\}$.

Lookback: pseudocosts [BGG+71]

Estimate for objective gain based on past branching observations.

- unit gain:
  computed from fractionalities f_j^* and LP gains
- pseudocosts Ψ_j^*:
  average unit gain of branching history
- branching decision based on estimated gains:

$$s(f_j^- \Psi_j^-, f_j^+ \Psi_j^+)$$



Select a fractional variable $j \in \underset{j' \in \mathcal{F}}{\operatorname{argmax}} \{s(f_j^- \Psi_j^-, f_j^+ \Psi_j^+)\}$.

Reliability branching [AKM04]

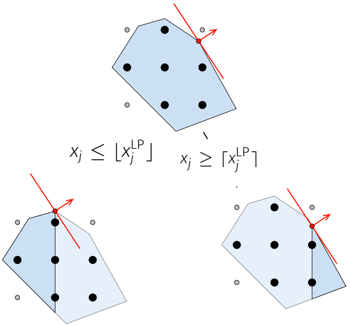
Pseudocosts are uninitialized at the beginning of the search.

Reliability branching

1. Determine the set of fractional variables $\mathcal{F} \neq \emptyset$.
2. Split $\mathcal{F}$ into reliable subset $\mathcal{F}^{\text{rel}}$ and unreliable subset $\mathcal{F}^{\text{url}}$.
3. Perform strong branching for all $j \in \mathcal{F}^{\text{url}}$.
4. Record unit gains and update pseudocosts.
5. Compare the best strong branching result with the best pseudocost prediction for the branching decision.

Variable branching information: other types

- cutoff information:
  average number of branchings yielding an infeasible node
- inference information:
  average number of domain reductions after branching
- conflict information:
  occurrence in recently learned conflict clauses
- conflict length information:
  occurrence in short conflicts



Hybrid branching [AB09]

Combine all types of variable branching history in a single, weighted score.

- scaling: divide each value by average over all variables
- normalize by $f\colon \mathbb{R}_{\geq 0} \to [0,1),\ x \mapsto \frac{x}{x+1}$
- use weights $\omega := (\omega^{\text{pscost}}, \omega^{\text{infer}}, \omega^{\text{prune}}, \omega^{\text{conf}}, \omega^{\text{clen}})$

Hybrid score

$$s_j := \omega * \left(f\left(\frac{s_j^{\text{pscost}}}{s_\varnothing^{\text{pscost}}}\right), f\left(\frac{s_j^{\text{infer}}}{s_\varnothing^{\text{infer}}}\right), f\left(\frac{s_j^{\text{prune}}}{s_\varnothing^{\text{prune}}}\right), f\left(\frac{s_j^{\text{conf}}}{s_\varnothing^{\text{conf}}}\right), f\left(\frac{s_j^{\text{clen}}}{s_\varnothing^{\text{clen}}}\right) \right)^T$$

SCIP implementation

$\omega^{\text{pscost}} = 1.0$, other weights $\in [10^{-4}, 10^{-2}]$

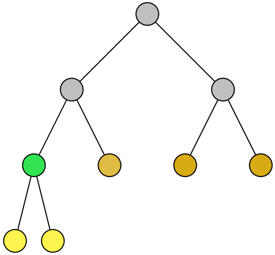
Other solvers may combine scores in a hierarchical or more complicated fashion.

Outline

Node Selection

Basic rules

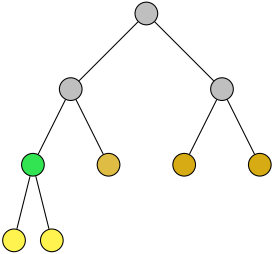
- depth first search (DFS)
  $\rightarrow$ keep simplex cost small
- best bound search (BBS)
  $\rightarrow$ improve dual bound
- best estimate search (BES)
  $\rightarrow$ improve primal bound



Node Selection

Basic rules

- depth first search (DFS)
  $\rightarrow$ keep simplex cost small
- best bound search (BBS)
  $\rightarrow$ improve dual bound
- best estimate search (BES)
  $\rightarrow$ improve primal bound



Best estimate [BGG+71]

Use learned pseudo costs to estimate objective value

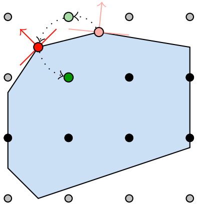
$$\hat{c} := c^T x_j^{\mathrm{LP}} + \sum_{j \in \mathcal{F}} \min\{f_j^- \Psi_j^-, f_j^+ \Psi_j^+\}$$

of the best solution in the subtree rooted at a node with LP solution x^{LP}.

Usually best bound/estimate interleaved with DFS plunges for simplex hot starting.

Outline

Primal Heuristics



Task

- improve primal bound
- effective on average
- guide remaining search

Techniques

- rounding
  - lock, randomized
  - octahedral neighborhood search
- diving
  - least infeasible
  - guided
  - solution density
- objective diving
  - objective feasibility pump
- large neighborhood search
  - RINS
  - RENS
  - local branching
- combinatorial
  - shift-and-propagate

RENS: Relaxation-Enforced Neighborhood Search [Ber14]

Input: MIP P

begin

$x^{\mathrm{LP}} \leftarrow$ relaxation optimum;

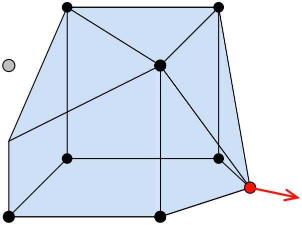
Fix all integral variables

$x_i := x_i^{\mathrm{LP}} \quad$ for all $i \in I : x_i^{\mathrm{LP}} \in \mathbb{Z}$;

Reduce domain of fractional variables

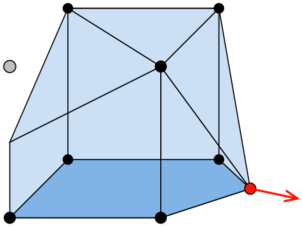
$x_i \in \{\lfloor x_i^{\mathrm{LP}} \rfloor ; \lceil x_i^{\mathrm{LP}} \rceil\}$;

Solve the resulting sub-MIP;



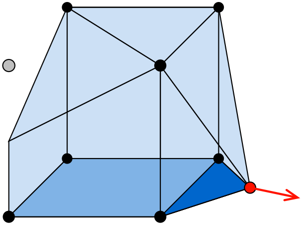
RENS: Relaxation-Enforced Neighborhood Search [Ber14]

Input: MIP P
begin
$x^{\mathrm{LP}} \leftarrow$ relaxation optimum;
Fix all integral variables
$x_i := x_i^{\mathrm{LP}}$ for all $i \in I : x_i^{\mathrm{LP}} \in \mathbb{Z}$;
Reduce domain of fractional variables
$x_i \in \{\lfloor x_i^{\mathrm{LP}} \rfloor; \lceil x_i^{\mathrm{LP}} \rceil\}$;
Solve the resulting sub-MIP;



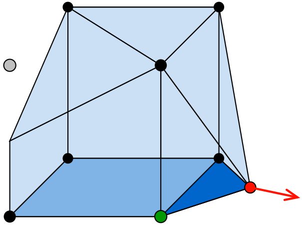
RENS: Relaxation-Enforced Neighborhood Search [Ber14]

Input: MIP P
begin
  $x^{\text{LP}} \leftarrow$ relaxation optimum;
  Fix all integral variables
  $x_i := x_i^{\text{LP}}$ for all $i \in I : x_i^{\text{LP}} \in \mathbb{Z}$;
  Reduce domain of fractional variables
  $x_i \in \{\lfloor x_i^{\text{LP}} \rfloor; \lceil x_i^{\text{LP}} \rceil\}$;
  Solve the resulting sub-MIP;



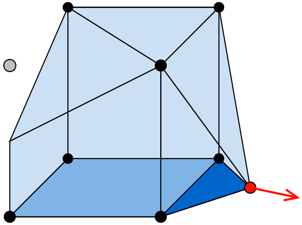
RENS: Relaxation-Enforced Neighborhood Search [Ber14]

Input: MIP P
begin
- $x^{\mathrm{LP}} \leftarrow$ relaxation optimum;
- Fix all integral variables
  $x_i := x_i^{\mathrm{LP}} \quad$ for all $i \in I : x_i^{\mathrm{LP}} \in \mathbb{Z}$;
- Reduce domain of fractional variables
  $x_i \in \{\lfloor x_i^{\mathrm{LP}} \rfloor ; \lceil x_i^{\mathrm{LP}} \rceil\}$;
- Solve the resulting sub-MIP;



RENS: Relaxation-Enforced Neighborhood Search [Ber14]

Input: MIP P
begin
  $x^{\mathrm{LP}} \leftarrow$ relaxation optimum;
  Fix all integral variables
  $x_i := x_i^{\mathrm{LP}} \quad$ for all $i \in I : x_i^{\mathrm{LP}} \in \mathbb{Z}$;
  Reduce domain of fractional variables
  $x_i \in \{\lfloor x_i^{\mathrm{LP}} \rfloor; \lceil x_i^{\mathrm{LP}} \rceil\}$;
  Solve the resulting sub-MIP;



RENS is only one example of many “fix-and-MIP” LNS heuristics.

Outline

Symmetry handling

Detection

- formulation symmetry either via graph automorphism [PR15] (in SCIP via bliss)
- or via dedicated symmetry detection on the constraint matrix (in many commercial solvers)

Symmetry handling

Detection

- formulation symmetry either via graph automorphism [PR15] (in SCIP via bliss)
- or via dedicated symmetry detection on the constraint matrix (in many commercial solvers)

Most common symmetry handling method: orbital fixing

- propagate symmetric variable fixings
- using local symmetry groups or stabilizers of 1-fixings

Symmetry handling

Detection

- formulation symmetry either via graph automorphism [PR15] (in SCIP via bliss)
- or via dedicated symmetry detection on the constraint matrix (in many commercial solvers)

Most common symmetry handling method: orbital fixing

- propagate symmetric variable fixings
- using local symmetry groups or stabilizers of 1-fixings

Alternatives:

- symmetry breaking constraints

$$\bar{c}x \geq \bar{c}\gamma(x),$$

  where $\bar{c} = (2^{n-1}, 2^{n-2}, \ldots, 2, 1) \in \mathbb{R}^n, x \in \{0,1\}^n$, or better their
- implicit enforcement via minimum cover inequalities [HP17], or recently
- derived from the Schreier-Sims table [Sal18].

Outline

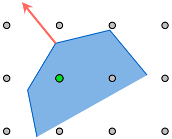
Numerics: Floating-point linear programming

Linear Program in standard form

$$\text{minimize} \;\; c^T x \;\; \text{subject to} \;\; Ax = b, x \geq \ell$$

Optimal solutions satisfy

- primal feasibility: $Ax - b = 0$ and $\hat{\ell}_i = \ell_i - x_i \leq 0$
- dual feasibility: $\hat{c}_i = c_i - y^T A_{\cdot i} \geq 0$
- complementary slackness: $\hat{c}_i \hat{\ell}_i = 0$



exact solution

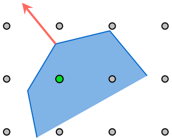
Numerics: Floating-point linear programming

Linear Program in standard form

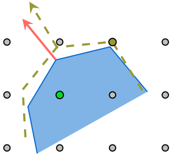
$$\text{minimize} \quad c^T x \quad \text{subject to} \quad Ax = b, x \geq \ell$$

Floating-point solvers compute solutions with residual errors

- primal feasibility: $\|Ax - b\|_\infty < \epsilon$ and $\hat{\ell}_i = \ell_i - x_i < \epsilon$
- dual feasibility: $\hat{c}_i = c_i - y^T A_{.i} > -\epsilon$
- complementary slackness: $|\hat{c}_i \hat{\ell}_i| < \epsilon$



exact solution



approximate solution

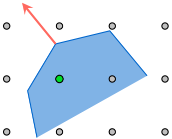
Numerics: Floating-point linear programming

Linear Program in standard form

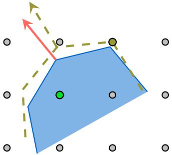
$$\text{minimize} \quad c^T x \quad \text{subject to} \quad Ax = b, x \geq \ell$$

Floating-point solvers compute solutions with residual errors

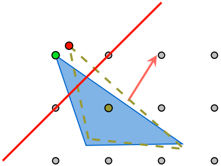
- primal feasibility: $\|Ax - b\|_\infty < \epsilon$ and $\hat{\ell}_i = \ell_i - x_i < \epsilon$
- dual feasibility: $\hat{c}_i = c_i - y^T A_{\cdot i} > -\epsilon$
- complementary slackness: $|\hat{c}_i \hat{\ell}_i| < \epsilon$



exact solution



approximate solution



invalid model strengthening

Exact Mixed-Integer Programming

Exact LP for

- primal solutions for fixed integer assignment
- dual bounding: expensive fallback
- hybrid solvers: QSopt_ex [ACDE07], SoPlex [GSW16]

Exact Mixed-Integer Programming

Exact LP for

- primal solutions for fixed integer assignment
- dual bounding: expensive fallback
- hybrid solvers: QSopt_ex [ACDE07], SoPlex [GSW16]

Fast and safe dual bounds

- hybrid arithmetic: floating-point– interval – rational
- correction of approximate dual solutions [NS04, SW13]

Exact Mixed-Integer Programming

Exact LP for

- primal solutions for fixed integer assignment
- dual bounding: expensive fallback
- hybrid solvers: QSopt_ex [ACDE07], SoPlex [GSW16]

Fast and safe dual bounds

- hybrid arithmetic: floating-point– interval – rational
- correction of approximate dual solutions [NS04, SW13]

Exact SCIP [CKSW13]

- exact LP-based branch-and-bound
- hierarchy of exact dual bounding techniques
  $\approx$ 2 – 4 times slower on numerically easy – difficult instances
- still gap to state-of-the-art MIP:
  no presolving, domain propagation, cutting planes, conflict analysis, heuristics

Outline

Motivation

Status quo: most MIP solvers do not output optimality certificates

Motivation

Status quo: most MIP solvers do not output optimality certificates

Difficulties

- practical: complex and diverse techniques in MIP solvers
- theoretical: no small certificates in general
- certificate form unclear: tree, cuts, superadditive function

Motivation

Status quo: most MIP solvers do not output optimality certificates

Difficulties

- practical: complex and diverse techniques in MIP solvers
- theoretical: no small certificates in general
- certificate form unclear: tree, cuts, superadditive function

Goals for a practical certificate format

- expressivity: encode all (or most) MIP techniques
- simplicity: checkable by small verification code
- generality: both more general and simpler than [ABC+09] for TSP

LP certificates

LP optimality can be certified by a dual solution

- e.g.

$$\begin{array}{rl} \min & 2x + y \\ \text{s.t.} & \\ C0: & 5x - y \geq 2 \\ C1: & 3x - 2y \leq 1 \end{array}$$

Given		
$C0:$	$5x - y \geq 2$	
$C1:$	$3x - 2y \leq 1$	
Derived		**Reason**
`obj` :	$2x + y \geq 1$	$\{1 \times C0 + (-1) \times C1\}$

LP certificates

LP optimality can be certified by a dual solution

- e.g.

$$\begin{array}{rl} \min & 2x + y \\ \text{s.t.} & \\ C0: & 5x - y \geq 2 \\ C1: & 3x - 2y \leq 1 \end{array}$$

Given		
$C0:$	$5x - y \geq 2$	
$C1:$	$3x - 2y \leq 1$	
Derived		**Reason**
`obj` :	$2x + y \geq 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
```

LP certificates

LP optimality can be certified by a dual solution

- e.g.

$$
\begin{array}{rl}
\min & 2x + y \\
\text{s.t.} & \\
C0: & 5x - y \geq 2 \\
C1: & 3x - 2y \leq 1
\end{array}
$$

Given		
$C0:$	$5x - y \geq 2$	
$C1:$	$3x - 2y \leq 1$	
Derived		**Reason**
`obj` :	$2x + y \geq 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
```

LP certificates

LP optimality can be certified by a dual solution

- e.g.

$$\begin{array}{rl} \min & 2x + y \\ \text{s.t.} & \\ C0: & 5x - y \geq 2 \\ C1: & 3x - 2y \leq 1 \end{array}$$

Given		
$C0:$	$5x - y \geq 2$	
$C1:$	$3x - 2y \leq 1$	
Derived		**Reason**
`obj` :	$2x + y \geq 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
RTP range 1 inf
```

LP certificates

LP optimality can be certified by a dual solution

- e.g.

$$\begin{array}{rl} \min & 2x + y \\ \text{s.t.} & \\ C0: & 5x - y \geq 2 \\ C1: & 3x - 2y \leq 1 \end{array}$$

Given		
$C0:$	$5x - y \geq 2$	
$C1:$	$3x - 2y \leq 1$	
Derived		**Reason**
obj :	$2x + y \geq 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
RTP range 1 inf
DER 1
  C2  G 1  2  0 2  1 1  { lin 2  0 1  1 -1 }
```

LP certificates

LP optimality can be certified by a primal-dual solution

- e.g.

$$\begin{array}{rl} \min & 2x + y \\ \text{s.t.} & \\ C0: & 5x - y \geq 2 \\ C1: & 3x - 2y \leq 1 \end{array}$$

Given		
$C0:$	$5x - y \geq 2$	
$C1:$	$3x - 2y \leq 1$	
Derived		**Reason**
`obj` :	$2x + y \geq 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2 x y
OBJ min 2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
RTP range 1 1
SOL 1
  2  0 3/7  1 1/7
DER 1
  C2  G 1  2  0 2  1 1  { lin 2  0 1  1 -1 }
```

Encoding Chvátal-Gomory cuts

Integer cutting planes often involve a rounding argument

- e.g.

$$\begin{array}{rl} \min & x+y \\ \text{s.t.} & \\ C0: & 4x+y \geq 1 \\ C1: & 4x-y \leq 2 \\ & x,y \in \mathbb{Z} \end{array}$$

Given		
	$x, y \in \mathbb{Z}$	
$C0:$	$4x + y \geq 1$	
$C1:$	$4x - y \leq 2$	
Derived		**Reason**
$C2:$	$y \geq -\frac{1}{2}$	$\{\frac{1}{2} \times C0 + (-\frac{1}{2}) \times C1\}$
$C3:$	$y \geq 0$	$\{\text{round up } C2\}$
$C4:$	$x + y \geq \frac{1}{4}$	$\{\frac{1}{4} \times C0 + \frac{3}{4} \times C3\}$
$C5:$	$x + y \geq 1$	$\{\text{round up } C4\}$

Encoding Chvátal-Gomory cuts

Integer cutting planes often involve a rounding argument

- e.g.

$$\begin{array}{rl} \min & x+y \\ \text{s.t.} & \\ C0: & 4x+y \geq 1 \\ C1: & 4x-y \leq 2 \\ & x,y \in \mathbb{Z} \end{array}$$

Given		
	$x, y \in \mathbb{Z}$	
$C0:$	$4x+y \geq 1$	
$C1:$	$4x-y \leq 2$	
Derived		**Reason**
$C2:$	$y \geq -\frac{1}{2}$	$\{\frac{1}{2} \times C0 + (-\frac{1}{2}) \times C1\}$
$C3:$	$y \geq 0$	$\{\text{round up } C2\}$
$C4:$	$x+y \geq \frac{1}{4}$	$\{\frac{1}{4} \times C0 + \frac{3}{4} \times C3\}$
$C5:$	$x+y \geq 1$	$\{\text{round up } C4\}$

- plain text syntax:

```
...
DER 4
  C2  G -1/2  1  1 1      { lin 2  0 1/2  1 -1/2 }
  C3  G 0   1  1 1        { rnd 2 }
  C4  G 1/4   2  0 1  1 1 { lin 2  0 1/4  3 3/4 }
  C5  G 1     2  0 1  1 1 { rnd 4 }
```

Encoding disjunctions

A tree-less branch-and-bound certificate

Given

$$x, y \in \mathbb{Z}$$

$$\begin{aligned} C0: &\quad 2x_1 + 3x_2 \geq 1 \\ C1: &\quad 3x_1 - 4x_2 \leq 2 \\ C2: &\quad -x_1 + 6x_2 \leq 3 \end{aligned}$$

Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leq 0$	$\{\text{assume}\}$	
$A1:$	$x_1 \geq 1$	$\{\text{assume}\}$	

Encoding disjunctions

A tree-less branch-and-bound certificate

Given

$$\begin{aligned} & x, y \in \mathbb{Z} \\ C0: \quad & 2x_1 + 3x_2 \geq 1 \\ C1: \quad & 3x_1 - 4x_2 \leq 2 \\ C2: \quad & -x_1 + 6x_2 \leq 3 \end{aligned}$$

Derived		Reason	Assumptions
$A0:$	$x_1 \leq 0$	$\{\text{assume}\}$	
$A1:$	$x_1 \geq 1$	$\{\text{assume}\}$	
$A2:$	$x_2 \leq 0$	$\{\text{assume}\}$	
$C3:$	$0 \geq 1$	$\{C0 + (-2) \times A0 + (-3) \times A2\}$	$A0, A2$

Encoding disjunctions

A tree-less branch-and-bound certificate

Given

$$x, y \in \mathbb{Z}$$

$$\begin{aligned} C0 &: \quad 2x_1 + 3x_2 \geq 1 \\ C1 &: \quad 3x_1 - 4x_2 \leq 2 \\ C2 &: \quad -x_1 + 6x_2 \leq 3 \end{aligned}$$

Derived		Reason	Assumptions
$A0$:	$x_1 \leq 0$	$\{\text{assume}\}$	
$A1$:	$x_1 \geq 1$	$\{\text{assume}\}$	
$A2$:	$x_2 \leq 0$	$\{\text{assume}\}$	
$C3$:	$0 \geq 1$	$\{C0 + (-2) \times A0 + (-3) \times A2\}$	$A0, A2$
$A3$:	$x_2 \geq 1$	$\{\text{assume}\}$	
$C4$:	$0 \geq 1$	$\{(-\frac{1}{3}) \times C2 + (-\frac{1}{3}) \times A0 + 2 \times A3\}$	$A0, A3$

Encoding disjunctions

A tree-less branch-and-bound certificate

Given

$$x, y \in \mathbb{Z}$$

$$\begin{aligned} C0 &: \quad 2x_1 + 3x_2 \geq 1 \\ C1 &: \quad 3x_1 - 4x_2 \leq 2 \\ C2 &: \quad -x_1 + 6x_2 \leq 3 \end{aligned}$$

Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leq 0$	$\{\text{assume}\}$	
$A1:$	$x_1 \geq 1$	$\{\text{assume}\}$	
$A2:$	$x_2 \leq 0$	$\{\text{assume}\}$	
$C3:$	$0 \geq 1$	$\{C0 + (-2) \times A0 + (-3) \times A2\}$	$A0, A2$
$A3:$	$x_2 \geq 1$	$\{\text{assume}\}$	
$C4:$	$0 \geq 1$	$\{(-\frac{1}{3}) \times C2 + (-\frac{1}{3}) \times A0 + 2 \times A3\}$	$A0, A3$
$C5:$	$0 \geq 1$	$\{\text{unsplit } C3, C4 \text{ on } A2, A3\}$	$A0$

Encoding disjunctions

A tree-less branch-and-bound certificate

Given

$$x, y \in \mathbb{Z}$$

$$\begin{aligned} C0: &\quad 2x_1 + 3x_2 \geq 1 \\ C1: &\quad 3x_1 - 4x_2 \leq 2 \\ C2: &\quad -x_1 + 6x_2 \leq 3 \end{aligned}$$

Derived		Reason	Assumptions
$A0:$	$x_1 \leq 0$	$\{\text{assume}\}$	
$A1:$	$x_1 \geq 1$	$\{\text{assume}\}$	
$A2:$	$x_2 \leq 0$	$\{\text{assume}\}$	
$C3:$	$0 \geq 1$	$\{C0 + (-2) \times A0 + (-3) \times A2\}$	$A0, A2$
$A3:$	$x_2 \geq 1$	$\{\text{assume}\}$	
$C4:$	$0 \geq 1$	$\{(-\frac{1}{3}) \times C2 + (-\frac{1}{3}) \times A0 + 2 \times A3\}$	$A0, A3$
$C5:$	$0 \geq 1$	$\{\text{unsplit } C3, C4 \text{ on } A2, A3\}$	$A0$
$C6:$	$x_2 \geq \frac{1}{4}$	$\{(-\frac{1}{4}) \times C1 + (\frac{3}{4}) \times A1\}$	$A1$
$C7:$	$x_2 \geq 1$	$\{\text{round up } C6\}$	$A1$
$C8:$	$0 \geq 1$	$\{(-\frac{1}{3}) \times C1 + (-1) \times C2 + \frac{14}{3} \times C7\}$	$A1$

Encoding disjunctions

A tree-less branch-and-bound certificate

Given

$$x, y \in \mathbb{Z}$$

$$\begin{aligned} C0:&\quad 2x_1 + 3x_2 \geq 1 \\ C1:&\quad 3x_1 - 4x_2 \leq 2 \\ C2:&\quad -x_1 + 6x_2 \leq 3 \end{aligned}$$

Derived		Reason	Assumptions
$A0:$	$x_1 \leq 0$	$\{\text{assume}\}$	
$A1:$	$x_1 \geq 1$	$\{\text{assume}\}$	
$A2:$	$x_2 \leq 0$	$\{\text{assume}\}$	
$C3:$	$0 \geq 1$	$\{C0 + (-2) \times A0 + (-3) \times A2\}$	$A0, A2$
$A3:$	$x_2 \geq 1$	$\{\text{assume}\}$	
$C4:$	$0 \geq 1$	$\{(-\frac{1}{3}) \times C2 + (-\frac{1}{3}) \times A0 + 2 \times A3\}$	$A0, A3$
$C5:$	$0 \geq 1$	$\{\text{unsplit } C3, C4 \text{ on } A2, A3\}$	$A0$
$C6:$	$x_2 \geq \frac{1}{4}$	$\{(-\frac{1}{4}) \times C1 + (\frac{3}{4}) \times A1\}$	$A1$
$C7:$	$x_2 \geq 1$	$\{\text{round up } C6\}$	$A1$
$C8:$	$0 \geq 1$	$\{(-\frac{1}{3}) \times C1 + (-1) \times C2 + \frac{14}{3} \times C7\}$	$A1$
$C9:$	$0 \geq 1$	$\{\text{unsplit } C5, C8 \text{ on } A0, A1\}$	$\emptyset$

VIPR: a straightforward branch-and-cut certificate [CGS17]

Simplicity

- only 4 reasoning types
- no explicit tree structure
- allows sequential checking

VIPR: a straightforward branch-and-cut certificate [CGS17]

Simplicity

- only 4 reasoning types
- no explicit tree structure
- allows sequential checking

Expressiveness

- split cuts

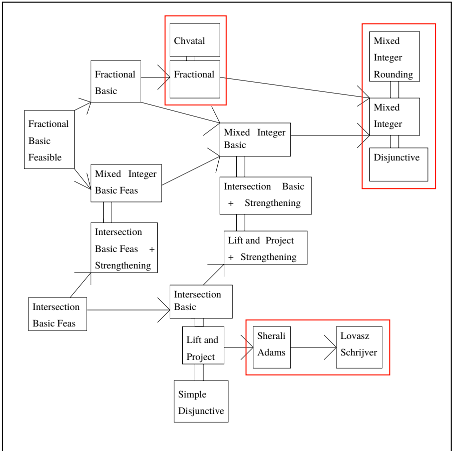


Figure 1: Comparison of Elementary Closures

Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

VIPR: a straightforward branch-and-cut certificate [CGS17]

Simplicity

- only 4 reasoning types
- no explicit tree structure
- allows sequential checking

Expressiveness

- split cuts
- constraint, reduced-cost propagation

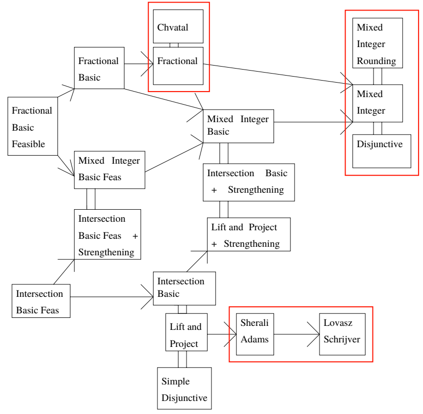


Figure 1: Comparison of Elementary Closures

Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

VIPR: a straightforward branch-and-cut certificate [CGS17]

Simplicity

- only 4 reasoning types
- no explicit tree structure
- allows sequential checking

Expressiveness

- split cuts
- constraint, reduced-cost propagation
- clique and implication graph, variable bound graph

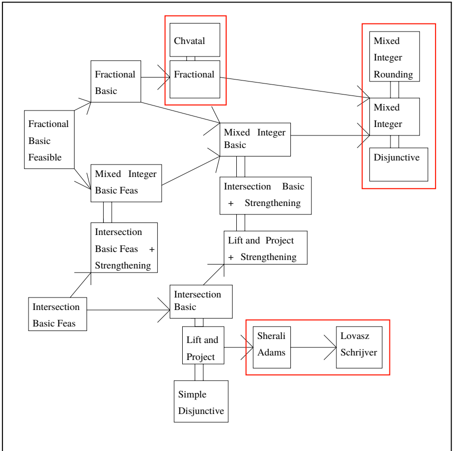


Figure 1: Comparison of Elementary Closures

Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

VIPR: a straightforward branch-and-cut certificate [CGS17]

Simplicity

- only 4 reasoning types
- no explicit tree structure
- allows sequential checking

Expressiveness

- split cuts
- constraint, reduced-cost propagation
- clique and implication graph, variable bound graph

Limitations

- certificate size
- translate MIP reasoning on-the-fly

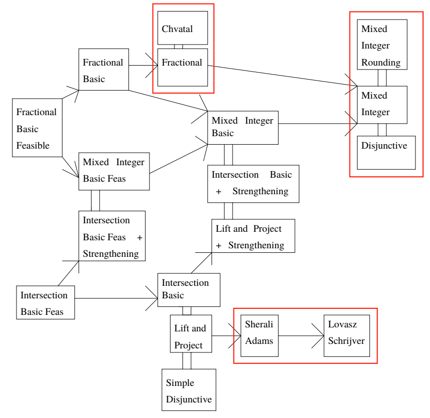


Figure 1: Comparison of Elementary Closures

Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

VIPR: a straightforward branch-and-cut certificate [CGS17]

Simplicity

- only 4 reasoning types
- no explicit tree structure
- allows sequential checking

Expressiveness

- split cuts
- constraint, reduced-cost propagation
- clique and implication graph, variable bound graph

Limitations

- certificate size
- translate MIP reasoning on-the-fly

Results

- Eifler, G, Pulaj 2018: Chvátal's Conjecture Holds for Ground Sets of Seven Elements, ZIB-Report 18-49 [EGP18]

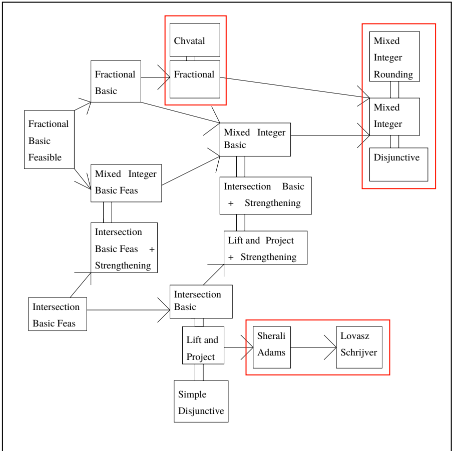


Figure 1: Comparison of Elementary Closures

Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

Wrap-up

Many things not covered:

- cut generation, selection, and management
- degeneracy and performance variability
- parallelization
- column generation and decomposition methods
- restarts
- benchmarking
- ...

Wrap-up

Many things not covered:

- cut generation, selection, and management
- degeneracy and performance variability
- parallelization
- column generation and decomposition methods
- restarts
- benchmarking
- …

Many questions…? *Thank you for your attention!*

References I

Tobias Achterberg and Timo Berthold.
Hybrid branching.
In Willem Jan van Hoeve and John N. Hooker, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 6th International Conference, CPAIOR 2009*, volume 5547 of *Lecture Notes in Computer Science*, pages 309–311. Springer, 2009.

David L. Applegate, Robert E. Bixby, VaÅ¡ek ChvÃ¡tal, William Cook, Daniel G. Espinoza, Marcos Goycoolea, and Keld Helsgaun.
Certification of an optimal tsp tour through 85,900 cities.
Operations Research Letters, 37(1):11 – 15, 2009.

David L. Applegate, William Cook, Sanjeeb Dash, and Daniel G. Espinoza.
Exact solutions to linear programming problems.
Operations Research Letters, 35(6):693–699, 2007.

Tobias Achterberg.
Conflict analysis in mixed integer programming.
Discrete Optimization, 4(1):4–20, 2007.

References II

Tobias Achterberg, Thorsten Koch, and Alexander Martin.
Branching rules revisited.
Operations Research Letters, 33(1):42–54, 2004.

Alper Atamtürk, George L Nemhauser, and Martin WP Savelsbergh.
Conflict graphs in solving integer programming problems.
European Journal of Operational Research, 121(1):40–55, 2000.

Timo Berthold.
RENS–the optimal rounding.
Mathematical Programming Computation, 6(1):33–54, 2014.

Michel Bénichou, Jean-Michel Gauthier, Paul Girodet, Gerard Hentges, Gerard Ribière, and O. Vincent.
Experiments in mixed-integer programming.
Mathematical Programming, 1:76–94, 1971.

Kevin KH Cheung, Ambros Gleixner, and Daniel E Steffy.
Verifying Integer Programming Results.
In *International Conference on Integer Programming and Combinatorial Optimization*, pages 148–160. Springer, 2017.

References III

William Cook, Thorsten Koch, Daniel E. Steffy, and Kati Wolter.
A hybrid branch-and-bound approach for exact rational mixed-integer programming.
Mathematical Programming Computation, 5(3):305–344, 2013.

R. J. Dakin.
A tree-search algorithm for mixed integer programming problems.
The Computer Journal, 8(3):250–255, 1965.

Leon Eifler, Ambros Gleixner, and Jonad Pulaj.
Chvátal's conjecture holds for ground sets of seven elements.
ZIB-Report 18-49, Zuse Institute Berlin, 2018.

Gerald Gamrath, Anna Melchiori, Timo Berthold, Ambros M. Gleixner, and Domenico Salvagnin.
Branching on multi-aggregated variables.
In *Integration of AI and OR Techniques in Constraint Programming*, volume 9075, pages 141–156, 2015.

Ralph E. Gomory.
Outline of an algorithm for integer solutions to linear programs.
Bulletin of the American Mathematical Society, 64(5):275–278, 1958.

References IV

Ambros M. Gleixner, Daniel E. Steffy, and Kati Wolter.
Iterative refinement for linear programming.
INFORMS Journal on Computing, 28(3):449–464, 2016.

Christopher Hojny and Marc E. Pfetsch.
Polytopes associated with symmetry handling.
Technical report, Technische Universität Darmstadt, 2017.

A. H. Land and A. G Doig.
An automatic method of solving discrete programming problems.
Econometrica, 28(3):497–520, 1960.

João P Marques-Silva and Karem Sakallah.
Grasp: A search algorithm for propositional satisfiability.
Computers, IEEE Transactions on, 48(5):506–521, 1999.

Arnold Neumaier and Oleg Shcherbina.
Safe bounds in linear and mixed-integer linear programming.
Mathematical Programming, 99(2):283–296, 2004.

References V

Marc E. Pfetsch and Thomas Rehn.
A computational comparison of symmetry handling methods for mixed integer programs.
Technical report, Optimization Online, 2015.

Domenico Salvagnin.
Symmetry breaking inequalities from the schreier-sims table.
In Willem-Jan van Hoeve, editor, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 521–529, Cham, 2018. Springer International Publishing.

Tuomas Sandholm and Robert Shields.
Nogood learning for mixed integer programming.
In *Workshop on Hybrid Methods and Branching Rules in Combinatorial Optimization, Montréal*, 2006.

Daniel E. Steffy and Kati Wolter.
Valid linear programming bounds for exact mixed-integer programming.
INFORMS Journal on Computing, 25(2):271–284, 2013.

References VI

Jakob Witzig, Timo Berthold, and Stefan Heinz.
Experiments with conflict analysis in mixed integer programming.
In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 211–220. Springer, 2017.