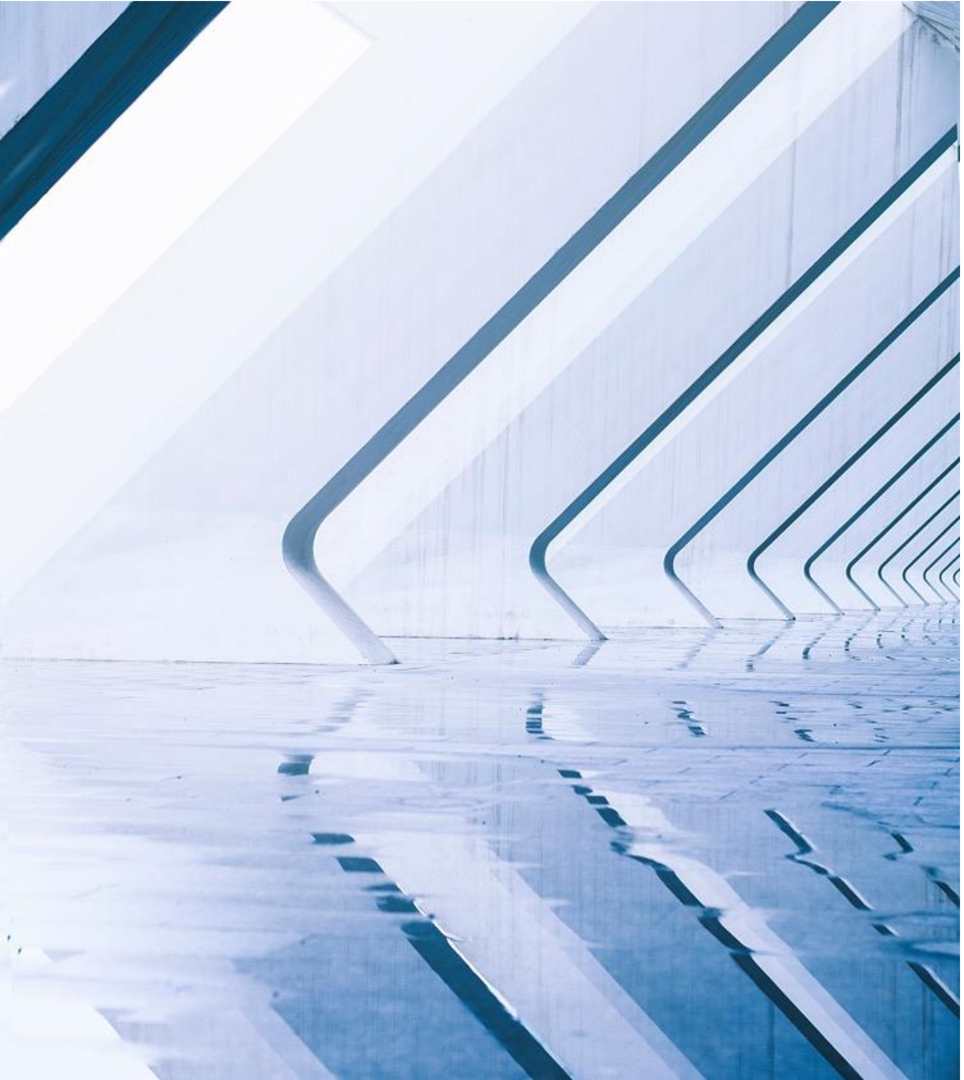


Presolving

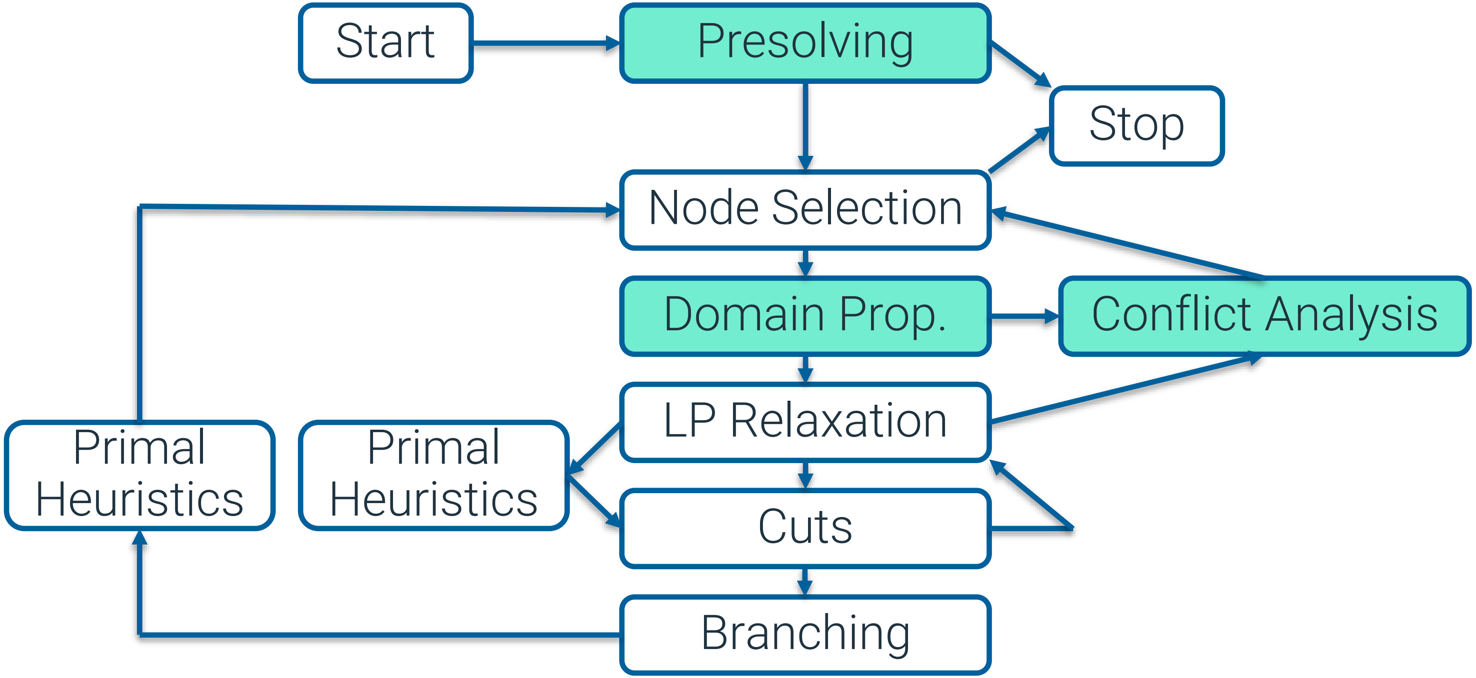
Strengthening model formulations a priori

Timo Berthold

Agenda

- Presolving
- Conflict Analysis
- Restarts

MIP Solver Flowchart



Presolving

LP presolving

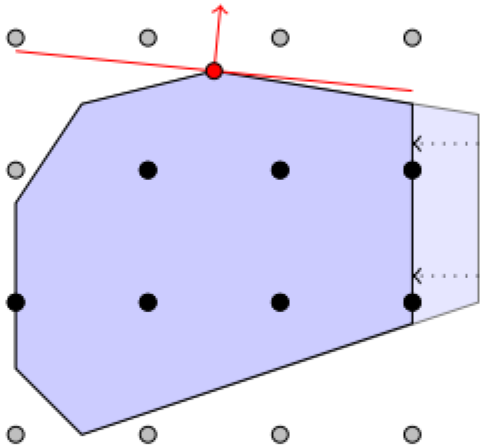
- Goals:
- Reduce problem size
  - Speed-up linear algebra during the solution process
- Strengthen LP relaxation
- Keep ability to postsolve primal & dual solutions and optimal basis
  - Preserve duality

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

- Dual Reductions:
  - consider objective function
  - at least one optimal solution remains

MIP presolving

- Like LP presolving, but more powerful:
- Exploit integrality
  - Round fractional bounds and right-hand sides
  - Lifting/coefficient strengthening
- Identify problem sub-structures
  - Cliques, implied bounds, networks, connected components, ...
- Does not need to preserve duality
  - We only need to be able to postsolve primal solutions



Trivial stuff

- Remove empty rows, columns
  - E.g., $0^T x \leq b_i < 0 \Rightarrow$ infeasible
- Tighten fractional bounds of integer variables
- Substitute fixed variables $x_j = c$ and aggregated variables $x_j = a x_k + c$
- Boundshifting of general integers: Replace $x_i \in \{N, N+1\}$ by binary variable
- Replace singleton rows
  - E.g., $a x_j \leq b_i, \mathrm{a} < 0 \Rightarrow x_j \geq \frac{b_i}{a} \Rightarrow$ new lower bound on x_j
- Normalize constraints
  - E.g., if all coefficients are integral, divide by greatest common divisor
- Upgrade constraints

Important!

- Problem instances are often automatically generated, contains many artifacts
- Often, the first modeling attempt is trivially infeasible or unbounded
  - Want to recognize this quickly
- Software that cannot recognize trivial things does not look trustworthy

Linear presolving

- Important concept: minimal and maximal activities (Brearly et al 1975)
- Let a linear constraint $a^T x \leq b$ and bounds $l \leq x \leq u$ be given.
  - $\alpha_{min} := \min\{a^T x : l \leq x \leq u\} = \sum_{j,a_j>0} a_j l_j + \sum_{j,a_j<0} a_j u_j$ is called minimal activity
  - $\alpha_{max} := \max\{a^T x : l \leq x \leq u\} = \sum_{j,a_j>0} a_j u_j + \sum_{j,a_j<0} a_j l_j$ is called maximal activity

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

Bound strengthening

- Important concept: minimal and maximal activities
- Let a linear constraint $a^T x \leq b$ and bounds $l \leq x \leq u$ be given.
  - $\alpha_{min} := \min\{a^T x : l \leq x \leq u\} = \sum_{j,a_j>0} a_j l_j + \sum_{j,a_j<0} a_j u_j$ is called minimal activity
  - $\alpha_{max} := \max\{a^T x : l \leq x \leq u\} = \sum_{j,a_j>0} a_j u_j + \sum_{j,a_j<0} a_j l_j$ is called maximal activity

- Second observation:
  - Let $a_i > 0$
  - $a^T x - a_i x_i + a_i x_i \leq b \Leftrightarrow x_i \leq \frac{b-(a^T x - a_i x_i)}{a_i} \quad \Rightarrow x_i \leq \frac{b-\alpha_{min}+a_i l_i}{a_i}$
  - For integer variables: $x_i \leq \left\lfloor \frac{b-\alpha_{min}+a_i l_i}{a_i} \right\rfloor$
  - Analogous for lower bound and max activity

Coefficient tightening

- Important concept: minimal and maximal activities
- Let a linear constraint $a^T x \leq b$ and bounds $l \leq x \leq u$ be given.
  - $\alpha_{min} := \min\{a^T x : l \leq x \leq u\} = \sum_{j,a_j>0} a_j l_j + \sum_{j,a_j<0} a_j u_j$ is called minimal activity
  - $\alpha_{max} := \max\{a^T x : l \leq x \leq u\} = \sum_{j,a_j>0} a_j u_j + \sum_{j,a_j<0} a_j l_j$ is called maximal activity

- Third observation:
  - Let $a_i > 0$, $x_i \in \{0,1\}$ and $a_{max} - a_i < b$
  - Then $a_i x_i + \sum_{j \neq i} a_j x_j \leq b$ can be reformulated as

$$(\alpha_{max} - b)x_i + \sum_{j \neq i} a_j x_j \leq (\alpha_{max} - a_i)$$

  - Proof: Check for 0 and 1. Again, other cases analogous

Example: Bound strengthening and coefficient tightening

- Example:
  - $7x_1 + 8x_2 \leq 21, 0 \leq x_1 \leq 3, 1 \leq x_2 \leq 3 \Rightarrow \alpha_{min} = 8$
    - $x_1 \leq \left\lfloor \frac{21-8+0}{7} \right\rfloor = 1$, $x_2 \leq \left\lfloor \frac{21-8+8}{8} \right\rfloor = 2$
    - Introduce $x_2 = 1 + y_2, y_2 \in \{0,1\}$
  - $7x_1 + 8y_2 \leq 13, 0 \leq x_1, y_2 \leq 1 \Rightarrow \alpha_{max} = 15$
    - $(15 - 13)\mathrm{x}_1 + 8\mathrm{y}_2 \leq 15 - 7 \Leftrightarrow 2x_1 + 8y_2 \leq 8$, $\alpha_{max} = 10$
    - $2x_1 + (10 - 8)y_2 \leq 10 - 8 \Leftrightarrow 2x_1 + 2y_2 \leq 2$
    - $x_1 + y_2 \leq 1$
  - That‘s a clique!

Bound strenghtening: little exercise

- Consider the linear inequality $3x_1 + 8x_2 + 5x_3 \leq 12$
  with integer variables $x_1 \geq 1, x_2 \geq 0$, $x_3 \geq -1, x \in \mathbb{Z}^3$
  - Compute the minimum activity of the constraint.
  - Use bound strengthening to compute upper bounds on all variables.

Bound strenghtening: little exercise

- Consider the linear inequality $3x_1 + 8x_2 + 5x_3 \leq 12$
  with integer variables $x_1 \geq 1, x_2 \geq 0$, $x_3 \geq -1, x \in \mathbb{Z}^3$
  - Compute the minimum activity of the constraint.
  - Use bound strengthening to compute upper bounds on all variables.

  - The minimum activity is $\alpha_{min} = 3 \cdot 1 + 0 + 5 \cdot (-1) = -2$.

Bound strenghtening: little exercise

- Consider the linear inequality $3x_1 + 8x_2 + 5x_3 \leq 12$
  with integer variables $x_1 \geq 1, x_2 \geq 0$, $x_3 \geq -1, x \in \mathbb{Z}^3$
  - Compute the minimum activity of the constraint.
  - Use bound strengthening to compute upper bounds on all variables.

  - The minimum activity is $\alpha_{min} = 3 \cdot 1 + 0 + 5 \cdot (-1) = -2$.
  - $x_1 \leq \left\lfloor \frac{12-(-2)+3}{3} \right\rfloor = 5$, $\quad x_2 \leq \left\lfloor \frac{12-(-2)+0}{8} \right\rfloor = 1$, $\quad x_3 \leq \left\lfloor \frac{12-(-2)-5}{5} \right\rfloor = 1$

Dual reductions

- Ensure that you keep at least one optimal solution
- Most simple example: Fixing variables that appear in no constraint
- Dual fixing: If variable x_j appears in no equation, only with nonnegative coefficients in $\leq$-constraints, with nonpositive coefficients in $\geq$-constraints and has a nonnegative objective, then x_j can be fixed to its lower bound
- Dual aggregation: Assume there is exactly one constraint violating the above, say x_j appears only in $\leq$-constraints, $c_i > 0$ and a_{ij} is its only negative coefficient.

  We can use $x_j = \frac{b_i}{a_{ij}} - \frac{1}{a_{ij}} \sum a_k x_k$
- Dual bound reduction: Strengthen bounds of variables to the tightest value for which all its constraints are redundant

Probing (Savelsbergh 1995)

- Strong branching without LP, only applying bound strengthening
- Usually only for binary variables, various working limits apply
- Sequence-dependent
- If $x_1 = 0 \Rightarrow$ infeasible and $x_1 = 1 \Rightarrow$ infeasible, then the problem is infeasible
- If $x_1 = 0 \Rightarrow$ infeasible, then fix $x_1 = 1$
- If $x_1 = 0 \Rightarrow x_2 = a$ and $x_1 = 1 \Rightarrow x_2 = b > a$, aggregate $x_2 = a + (b - a)x_1$
- If $x_1 = 0 \Rightarrow x_2 \leq a$ and $x_1 = 1 \Rightarrow x_2 \leq b$, then apply $x_2 \leq \max(a, b)$
- If $x_1 = 0 \Rightarrow x_2 \leq a$, store information in implication graph, use for heuristics, lifting, ...

OBBT (e.g., Gleixner et al 2017)

- Optimization-based bound tightening
  - Maximize/minimize each variable over $Ax \leq b$
  - Can exploit integrality for primal reductions, but not for dual reductions
  - Gives tightest possible bound
- Typically not employed in MIP solving, but important technique in MINLP

Multi-Row/Column reductions

- Parallel rows/columns
  - Search for pairs of rows such that coefficient vectors are parallel to each other
    - Hashing plus sorting algorithm
  - Discard the dominated row, or merge two inequalities into an equation
- Dominated rows/columns
  - Pairwise comparison, heuristic selection of pairs
- Sparsification
  - Add equations to other rows in order to cancel non-zeros
- Clique merging:
  - Merge multiple cliques into one larger clique:
  - $x_1 + x_2 \leq 1, x_2 + x_3 \leq 1, x_1 + x_3 \leq 1 \Rightarrow x_1 + x_2 + x_3 \leq 1$

Many more…

- Implied integer detection
  - $\sum a_j x_j + y = b$, x_j integer variables $a_j \in \mathbb{Z}\ \forall j$ and $b \in \mathbb{Z}$, then y integer
- GCD reduction
  - Let gcd be the GCD of all coefficients a_j in a row
  - $\sum \frac{a_j}{gcd} x_j \leq \left\lfloor \frac{b}{gcd} \right\rfloor$
- Disconnected component detection
  - DFS on matrix A
  - If there is an independent component $\tilde{A}\tilde{x} \leq \tilde{b}$ that is not connected to the rest of $Ax \leq b$, solve it as auxiliary MIP (if it is small enough)
- Analytic center presolving (Berthold et al 2017)
  - Fix variables that are at one of their bounds in the analytic center

Reduced cost fixing

- Not yet a cut, not presolving anymore
- Reduced costs: $\mathrm{r} \coloneqq c - A^T y$ for optimal dual solution y
  - Zero for basic variables, nonnegative for nonbasic variables at lower bound, nonpositive for nonbasic variables at upper bound
- Unit amount by which LP value would change, if we shifted solution towards other bound
  - Can be used to tighten bounds of integer variable
  - For binary $\mathrm{x_i}$ and LB the dual bound, UB the primal bound of an optimization problem, fix $x_i = 0$, if $r_i \geq UB - LB$.
- Apply locally with current LP solution
- Globally, store best reduced cost per variable from any global LP optimum (cut loop!)
  - Reconsider every time when UB changes

Quiz time

- Presolving
  a) Must not cut off any feasible solution
  b) May cut off feasible, but must not cut off optimal solutions
  c) May cut off optimal solutions
- The maximum activity of $x_1 - x_2 + 2x_3 \leq 5$, $x_1, x_2, x_3 \in \{1,2\}$ is
  a) 2
  b) 5
  c) 6
- Which of the following is not a goal of presolving?
  a) Shrink the problem size
  b) Find an initial solution
  c) Strenghten the LP relaxation

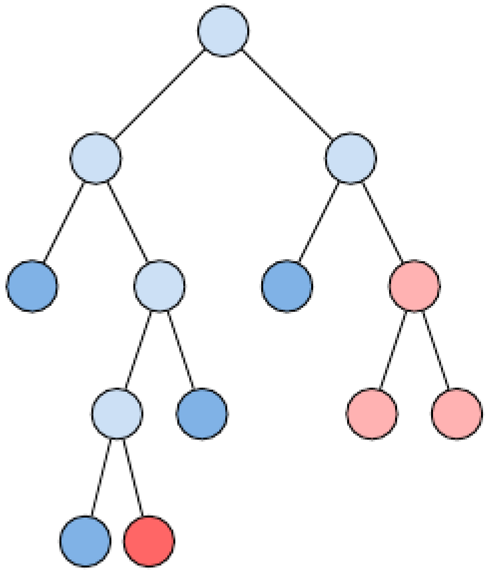
Quiz time

- Presolving
  a) Must not cut off any feasible solution
  b) May cut off feasible, but must not cut off optimal solutions
  c) May cut off optimal solutions
- The maximum activity of $x_1 - x_2 + 2x_3 \leq 5, x_1, x_2, x_3 \in \{1,2\}$ is
  a) 2
  b) 5
  c) 6
- Which of the following is not a goal of presolving?
  a) Shrink the problem size
  b) Find an initial solution
  c) Strenghten the LP relaxation

Conflict analysis

Conflict analysis: Motivation

- Half of the nodes in a binary B&B tree are pruned
  - Infeasibility, Bound-exceeding, solution
- Try to learn what led to infeasibility
  - Generate valid constraints
  - Cut off other parts of the tree
  - Use for propagation
- Sources of infeasibility:
  - Propagation (node presolve)
  - Infeasible LP



Idea comes from SAT solving (Moskewicz et al 2001)

- Boolean variables $x_1, \dots, x_n \in \{0,1\}$
- Clause $C_i: l_{i1} \vee \cdots \vee l_{ik}$ with literals $l_{ij} = x_j$ or $l_{ij} = \bar{x}_j = 1 - x_j$
- Find assignment that SATisifies all clauses or prove that no such assigment exists
  - THE NP-complete problem
  - Trivial to reformulate as binary MIP without objective
- Working horse unit propagation:
  - All but one literal fixed ⇒ last literal
  - E.g., clause: $x_6 \vee \bar{x}_7 \vee x_9 \vee x_{10}$
  - Fixings: $x_6 = 0, x_7 = 1, x_9 = 0$
  - Deduction: $\bar{x}_6 \wedge x_7 \wedge \bar{x}_9 \Rightarrow x_{10}$

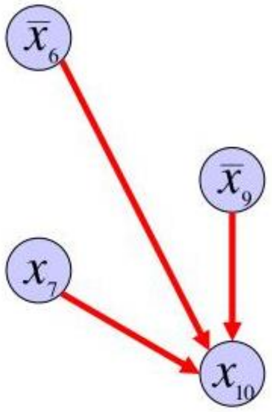


image source: Tobias Achterberg

The conflict graph

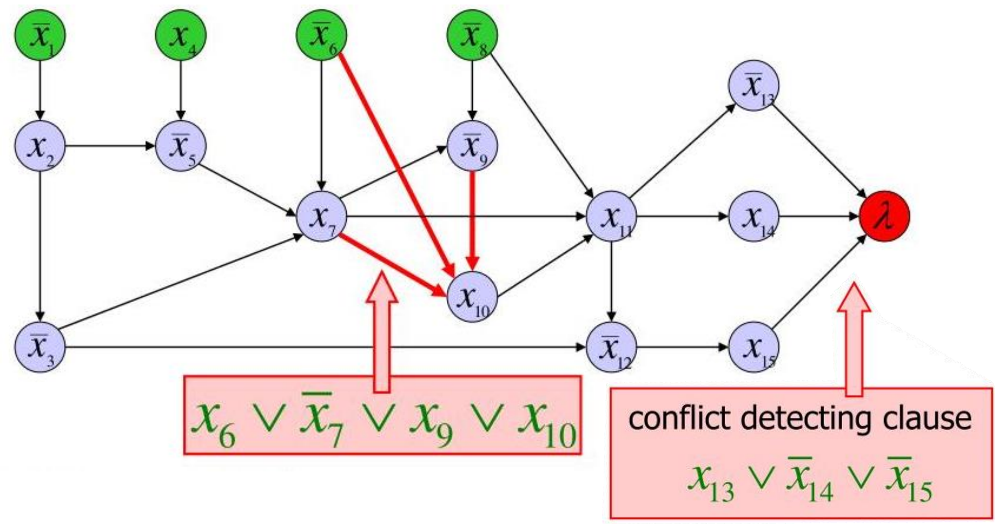
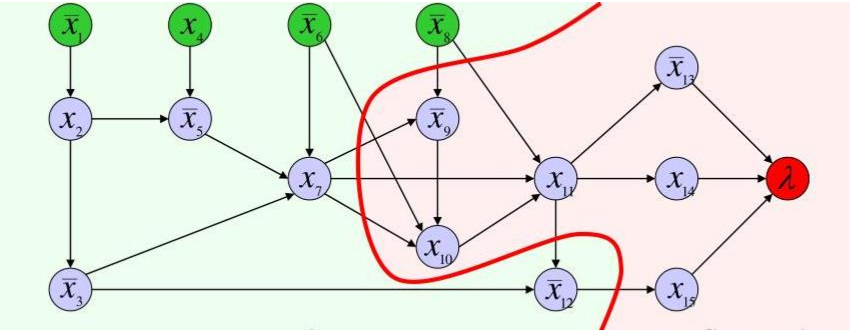


image source: Tobias Achterberg

- Graph capturing the ensemble of logic deductions that led to the current state (infeasible)
  - Nodes represent variable fixings, ingoing arcs represent a reason for a deduction
  - Green nodes: branching decisions, blue nodes: deduced fixings, red node: infeasibility

Cuts lead to cuts

- Every cut that separates branching decisions from conflict vertex gives rise to a conflict constraint
  - $x_6 \vee \bar{x}_7 \vee x_8 \vee x_{12}$
- Trivial cuts:
  - λ-cut: $x_{13} \vee \bar{x}_{14} \vee \bar{x}_{15}$
    - We already knew that…
  - No-good-cut/decision-cut:
    $x_1 \vee \bar{x}_4 \vee x_6 \vee x_8$
    - Good, if we start from scratch
    - Otherwise, we will never use it



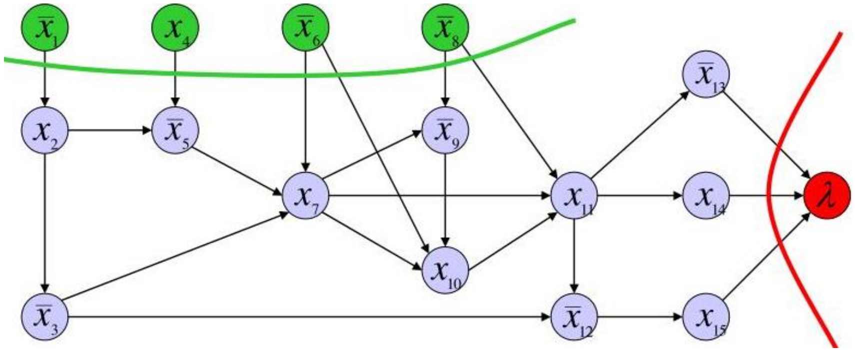


image source: Tobias Achterberg

First unique implication points

- Unique implication point: Lies on all paths from the last decision vertex to the conflict vertex
- First-UIP: the UIP closest to the conflict vertex (here: x_{11})
- First-UIP-Cut: everything fixed after First UIP on conflict side. Here: $x_3 \vee \bar{x}_{11}$
- SAT-solver typically use 1-FUIP cuts, MIP solvers All-FUIP

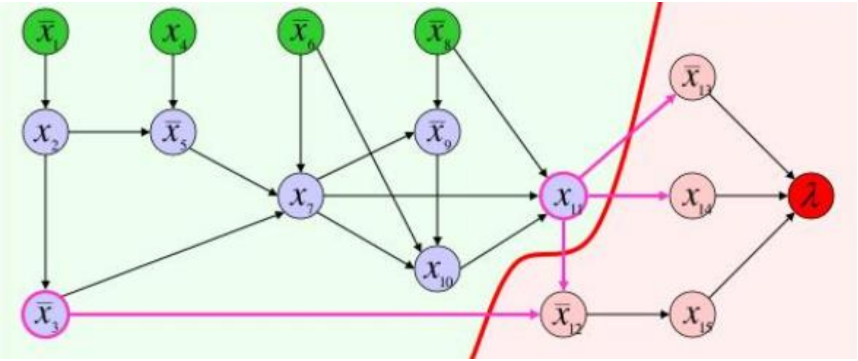


image source: Tobias Achterberg

Conflict Analysis in MIP (Achterberg 2007)

- Non-binary variables
  - Use bound changes instead of fixings
  - Relax literals with continuous variables by including equality
- Infeasibility often detected by LP
  - All local bound changes lead to Infeasibility???
  - Try to find subset that still proves infeasibility
    - Non-zeros in dual ray
    - Greedily sparsify dual ray
    - Start conflict analysis from those bound changes

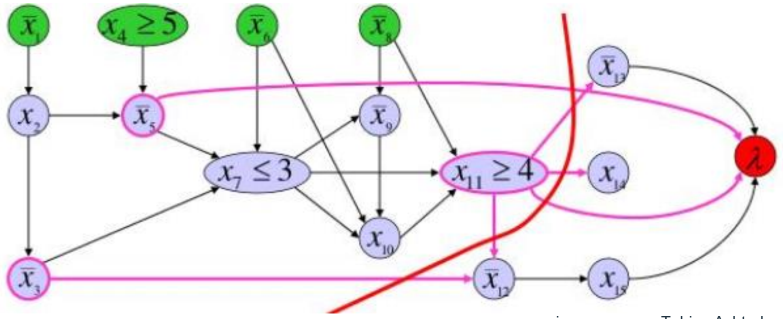


image source: Tobias Achterberg

Conflict Analysis: Implementation details

- Create conflict cuts from:
  - Propagation conflicts
  - Infeasible LPs
  - Bound-exceeding LPs
  - Diving heuristic LPs
  - Strong branching LPs
  - Integer feasible LPs (!)
- Use conflict constraints only for propagation
- Use aging or pooling mechanism to maintain short list of conflicts

Rapid Learning (Berthold et al 2010)

- MIP solvers spent a lot of time in root node processing
  - Exhaustive presolving algorithms (probing, dominated columns)
  - Initial LP solve often takes 100-1000x simplex iterations
  - Cutting plane generation
  - LNS heuristics (sub-MIPs)
  - Strong branching
- Idea: SAT-style search (propagation&conflicts, no LP)
  - Will take a fraction of the root time
  - Provides heaps of global information
    - Conflicts, global bound changes
    - Branching statistics
    - Feasible solutions, might even solve the problem

Restarts

Restarts

- Common practice in SAT solving
  - Periodically (with exponential increase) restart solve
  - Use learnt conflicts to steer search into better direction
  - Tailored towards feasibility problems
    - Need to get lucky once
- Motivation in MIP: presolving
  - Many procedures that are only applied in presolving
  - Global fixings found later might lead to further reductions
  - Tighter presolved problem leads to better cuts, primal solutions, …
- Question: How can we detect a good point to restart?

Restart reasoning

- Restart at root node
  - When many variables have been fixed
  - When variables with high impact have been fixed
  - When optimization problem turned into feasibility problem
- Restart during tree (Anderson et al 2018)
  - Can be used to change branching (and cutting, heuristic,...) strategy
    - Tailored towards easy instance before,
      hard instances after restarts
  - Challenge: Need to predict if the search will
    last longer anyway or is about to finish

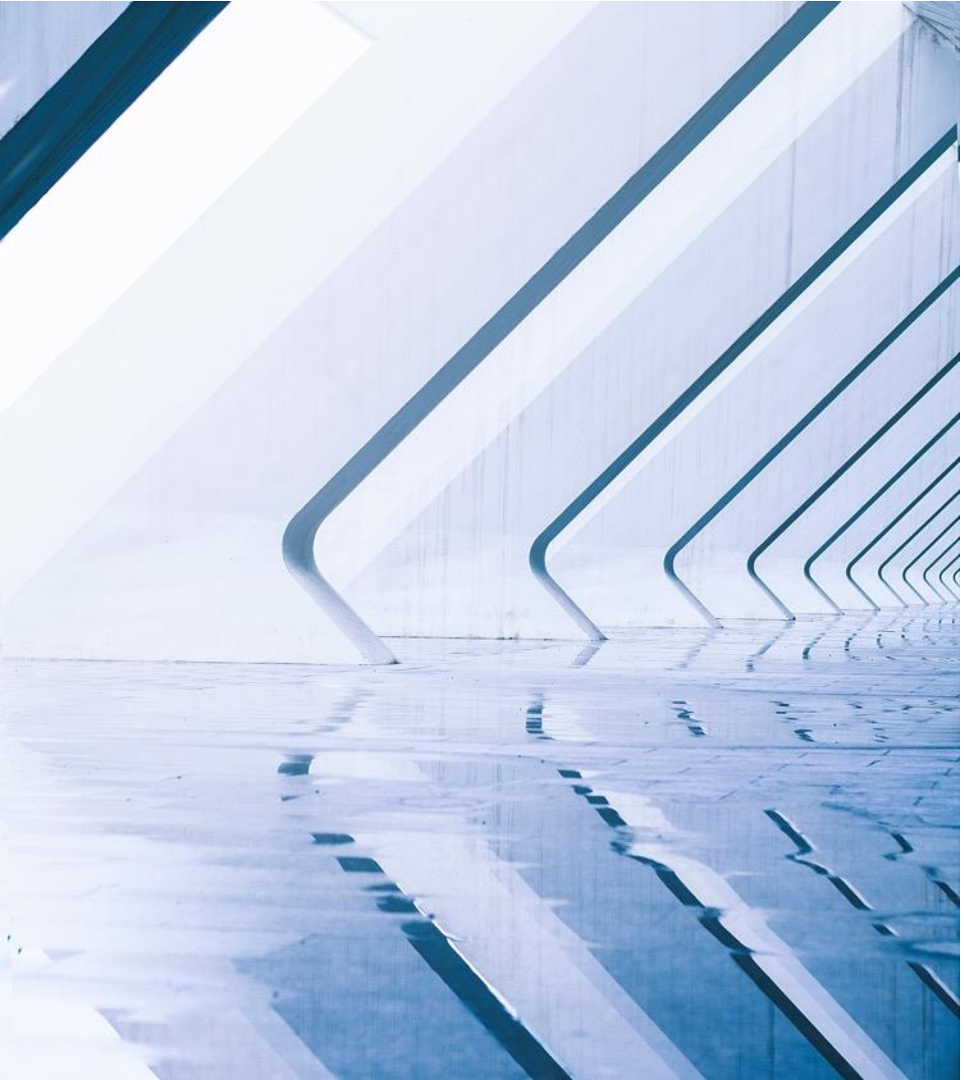
Quiz time

- A conflict constraint is a set of variable bounds from which
  a) At least one has to hold in any feasible solution
  b) At most one can hold in any feasible solution
  c) All but one have to hold in any feasible solution
- Conflict analysis for infeasible LPs uses
  a) A primal solution
  b) A dual ray
  c) The reduced costs
- Where do MIP solvers NOT restart?
  a) During the initial LP solve
  b) During the cut loop
  c) During the tree search

Quiz time

- A conflict constraint is a set of variable bounds from which
  a) At least one has to hold in any feasible solution
  b) At most one can hold in any feasible solution
  c) All but one have to hold in any feasible solution
- Conflict analysis for infeasible LPs uses
  a) A primal solution
  b) A dual ray
  c) The reduced costs
- Where do MIP solvers NOT restart?
  a) During the initial LP solve
  b) During the cut loop
  c) During the tree search

FICO



FICO

Thank You!

Timo Berthold