MIP heuristics in commercial solvers, part II

Pietro Belotti
Xpress Solver Development, FICO

Outline

- Primal Dual Integral
- Heuristics in FICO-Xpress
- Local search: good neighborhoods
- Heuristic based on analytic center

How do we measure the added value of a primal heuristic?

- Time to optimality t_{solved} (or # BB nodes)
  - very much depends on dual bound
- Time to first solution t_1
  - disregards solution quality
- Time to best solution t_{opt}
  - nearly optimal solution might be found long before

We would like to assess the *impact* of a heuristic on the overall solve process.

Primal integral

Suppose x_{opt} is the optimal solution and the time limit is t_{max}.

Def.: the primal gap w.r.t. a solution $\tilde{x}$, defined as $\gamma(\tilde{x}) \in [0,1]$, is

$$\gamma(\tilde{x}) = \begin{cases} 0 & \text{if } c^T x_{opt} = c^T \tilde{x} \\ 1 & \text{if } c^T x_{opt} \cdot c^T \tilde{x} < 0 \\ \dfrac{|c^T(x_{opt} - \tilde{x})|}{\max(|c^T x_{opt}|, |c^T \tilde{x}|)} & \text{otherwise.} \end{cases}$$

If $\tilde{x}(t)$ is the incumbent at time t, the primal gap function $p: [0, t_{max}] \rightarrow [0,1]$ is

$$p(t) = \begin{cases} 1 & \text{if no incumbent at } t \\ \gamma(\tilde{x}(t)) & \text{otherwise.} \end{cases}$$

Primal integral

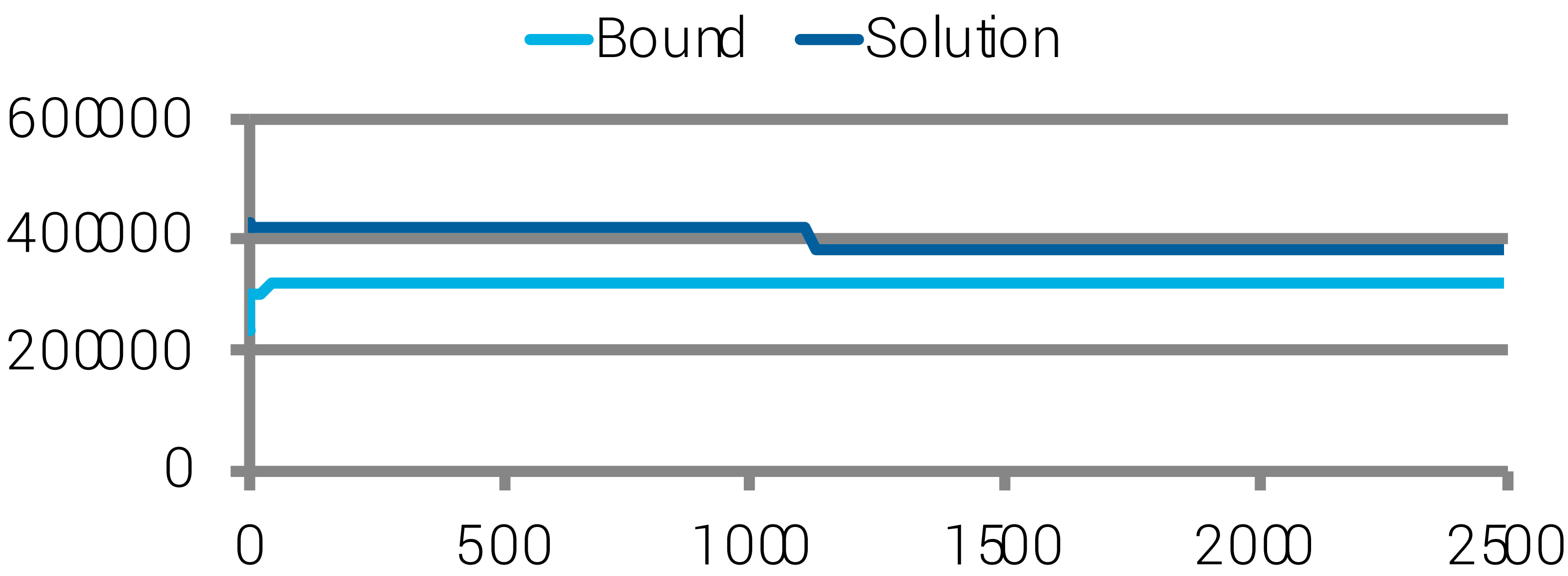
Integrating $p(t)$ over $[0, T]$ for $\mathrm{T} \in [0, t_{max}]$ yields a measure of the heuristic:

$$P(T) = \int_0^T p(t)dt = \sum_{i=1}^{I} p(t_{i-1}) \cdot (t_i - t_{i-1})$$

- $P(t_{max}) \approx 0$: good solutions were found early in the solution process
- $P(t_{max}) \approx t_{max}$: solutions were either not found early or they were poor.
- It favors finding good solutions early
- Considers each update of incumbent
- $\frac{P(t_{max})}{t_{max}}$ is an indicator of the "average solution quality"
- We get the expected quality of the incumbent in case of timeout

Primal Dual Integral

- Includes the dual (lower for min. problems) bound in the measure
- Highlights the influence of the heuristics on the overall solve process
- Useful when optimal solution not known



Sequence of heuristics in FICO-Xpress

1. Presole
2. Run heuristic
3. LP solve
4. Select diving strategy by running different types
5. Cut + heuristic loop (diving, possibly local search)
6. Reconsider diving strategy, run all and select one to be run in the tree
7. Run heuristic: Local search RINS + MIP/LP-centered (aka "proximity search")
8. BB tree:
   1. RINS, diving, rounding-based heuristics

Classes of heuristics

The workforce is broadly divided in

- “Diving”: really means a combination of
  - Rounding
  - Fix + propagate
  - Diving
- Local search: Large Neighborhood Search or Variable Neighborhood search

Heuristics in Xpress

4 Rounding/simple heuristics

10 Diving heuristics

3 Structural heuristics

2 Feasibility Pumps

4 Local search heuristics

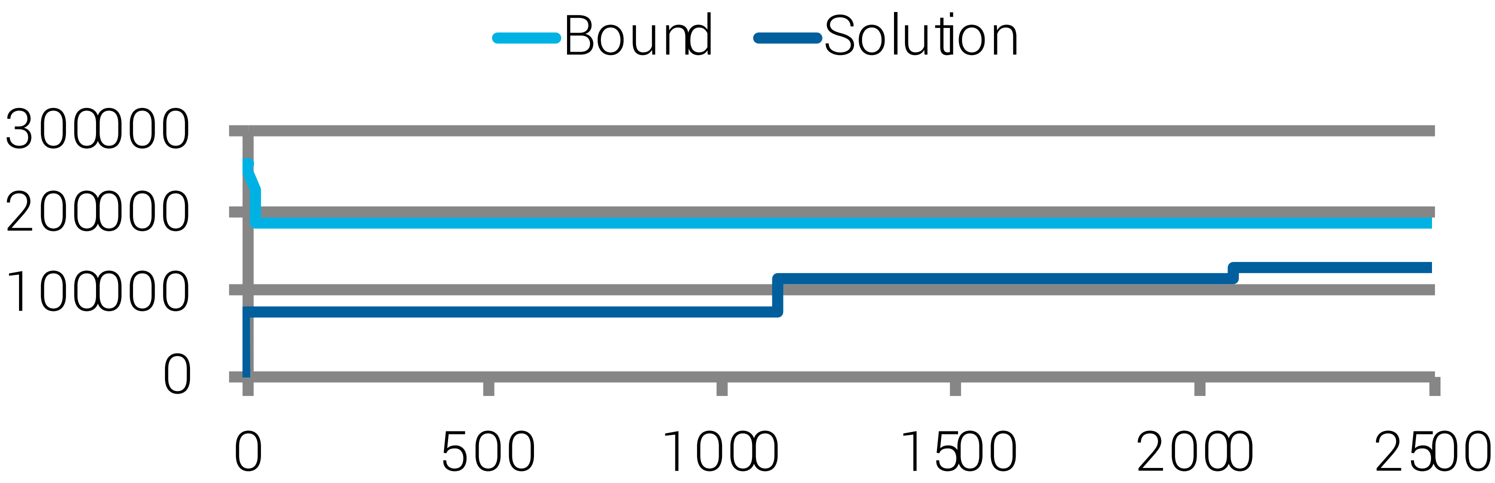
1 User induced local search heuristic

Many heuristics are off by default → not sufficient benefit when solving to optimality.

Local search heuristics

- Problem is too large to wait for a single dive to complete
- Correct bad branching choices
- A good starting solution can benefit the branch-and-bound search
  - Useful for some special problem structures:
    - No duality gap, so can stop when an optimal solution is found (“lucky” heuristics)
    - When a good heuristic solution leads to lots of ready cost fixings and therefore a significant reduction in the problem
  - 25% slowdown in time to optimality on internal MIP benchmark set when switching all heuristics off

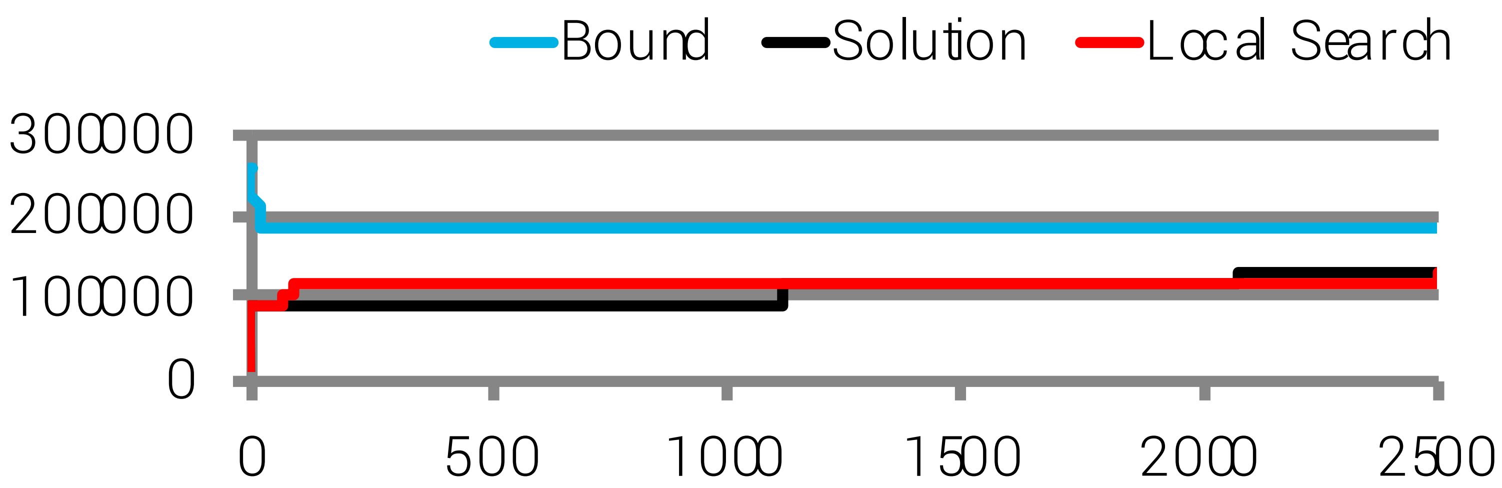
Benefit of Local Search Heuristics



Hard user problem [*maximization*].
Initial improvement in bound from cutting

Initial solution from simple heuristics, but better solutions found only through diving (~1000s per dive)

Benefit of Local Search Heuristics



Local search heuristic can improve initial diving heuristic solution.
Same quality solution with 50 sec. local search as 1000 sec. dive.

Basics of a Local Search Heuristics

- Given an existing MIP solution, x^*
  - Feasibility not required
  - Can be provided by a constructive heuristic (e.g. diving)
- Select one or more *critical* variables $C' \subseteq C^*$.
  C^*: subset of integer variables $x_j, j \in I$, such that x can be an improving solution only if $x_j \neq x_j^*$ for some $j \in C^*$
- Select a subset of variables $J' \subset J$, with $C' \subseteq J'$
- Solve the induced local search MIP by fixing all variables not in J' to x^*:

$$
\begin{array}{ll}
\min & cx \\
\text{s.t.} & Ax \leq b \\
& x_j = x_j^*, \forall j \in J \setminus J' \\
& x_j \in \mathbb{Z}, \forall j \in I
\end{array}
$$

Finding Critical Variables

- Given MIP solution x^*, fix integer variables and solve

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & l_j \leq x_j \leq u_j, \forall j \in J \\ & x_j = x_j^*, \forall j \in I \end{aligned}$$

- Use reduced costs

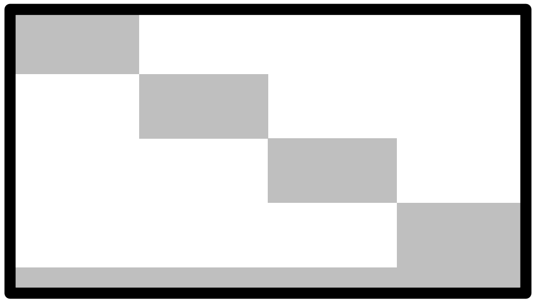
$$r_j = c_j - c_B^T B^{-1} A_j, \forall j \in J$$

- Change $x^* \rightarrow x^* + \Delta x$ has approximate cost change $r \cdot \Delta x$
- $j \in I$ critical for x^* iff

$$r_j > 0, x_j^* > l_j \text{ or}$$
$$r_j < 0, x_j^* < u_j$$

Neighborhood Selection

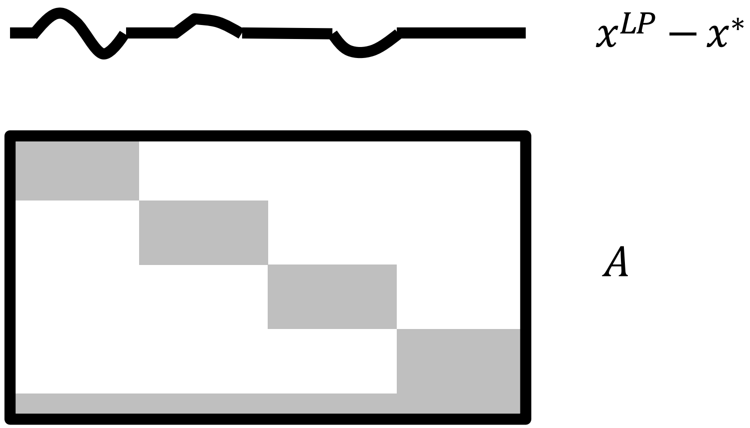
- Use an LP solution x^{LP}.
- Select subset J' as set of variables where x^{LP} and x^* differs.



A

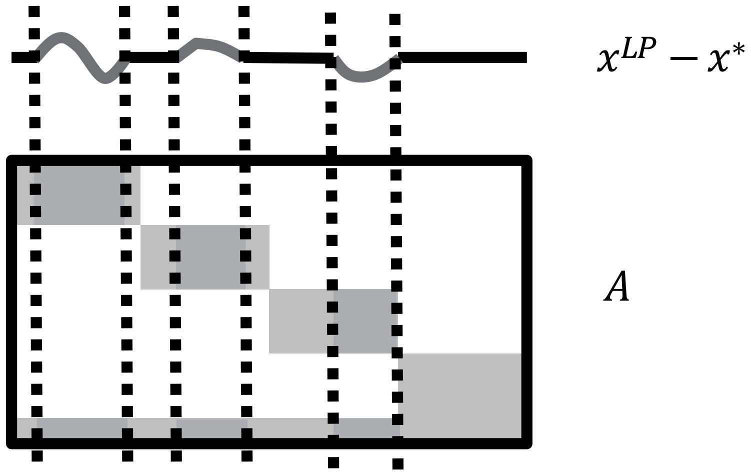
Neighborhood Selection

- Use an LP solution x^{LP}.
- Select subset J' as set of variables where x^{LP} and x^* differs.



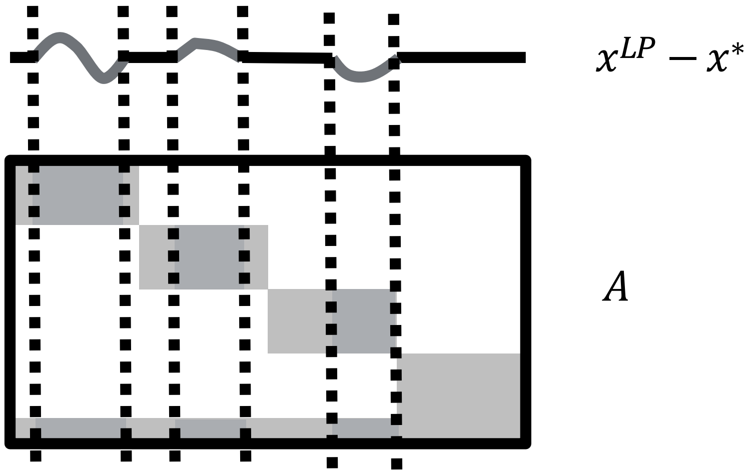
Neighborhood Selection

- Use an LP solution x^{LP}.
- Select subset J' as set of variables where x^{LP} and x^* differs.



Neighborhood Selection

- Use an LP solution x^{LP}.
- Select subset J' as set of variables where x^{LP} and x^* differs.



- RINS (Relaxation Induced Neighborhood Search), *Danna, Rothberg, Le Pape (2005)*
- In practice, increase neighborhood to get an appropriately sized MIP.

Neighborhood Selection (2)

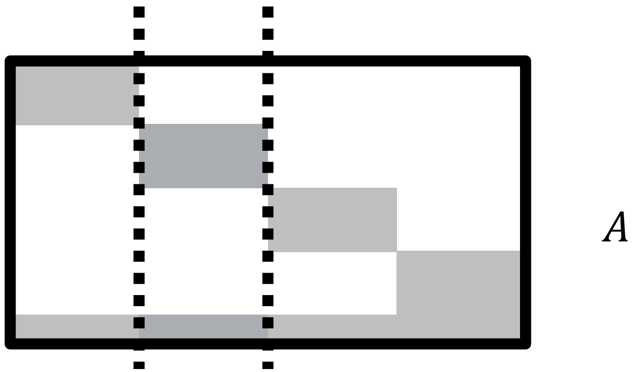
- For structured problems, look for related variables, e.g.:
  - Blocks of block-angular structure (stochastic models)
  - Time intervals for time-period formulations (unit commitment, lot sizing)

1. Select a random block or time interval
2. Re-optimize induced MIP
3. Repeat

Neighborhood Selection (2)

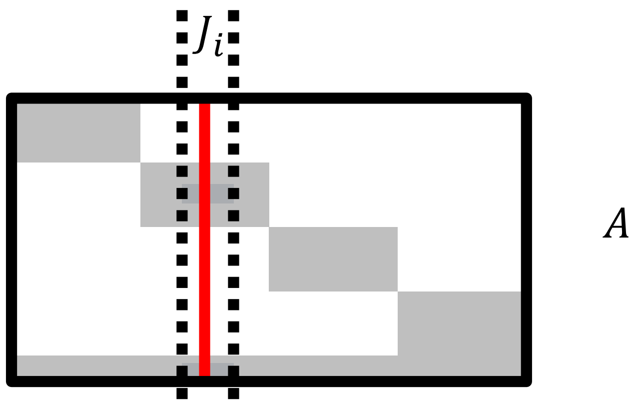
- For structured problems, look for related variables, e.g.:
  - Blocks of block-angular structure (stochastic models)
  - Time intervals for time-period formulations (unit commitment, lot sizing)

1. Select a random block or time interval
2. Re-optimize induced MIP
3. Repeat



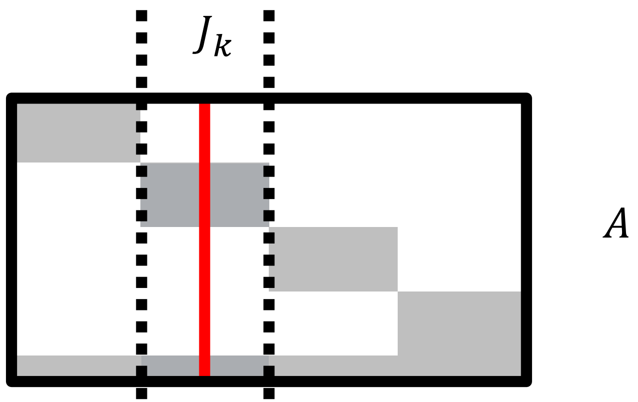
Building a Nice Neighborhood

- Create an initial neighborhood $J_0 \subseteq C^*$ containing one or more critical variables for x^*.
- Incrementally augment J_0 with variables closely connected to those in J_0.
- Alternatively, rank variables $j \in J \setminus J_0$ based on connectivity to J_0 and exclude least connected variables.



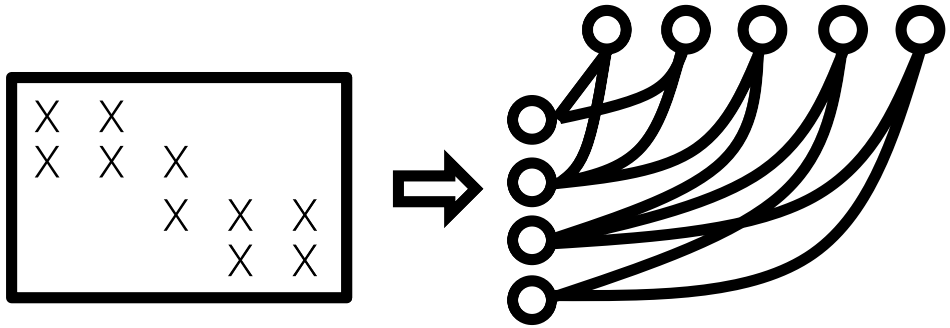
Building a Nice Neighborhood

- Create an initial neighborhood $J_0 \subseteq C^*$ containing one or more critical variables for x^*.
- Incrementally augment J_0 with variables closely connected to those in J_0.
- Alternatively, rank variables $j \in J \setminus J_0$ based on connectivity to J_0 and exclude least connected variables.



Building a Nice Neighborhood

- Translate variable relatedness problem into a graph connectivity problem on a bipartite graph:



- Xpress uses both the weighted and unweighted graphs.

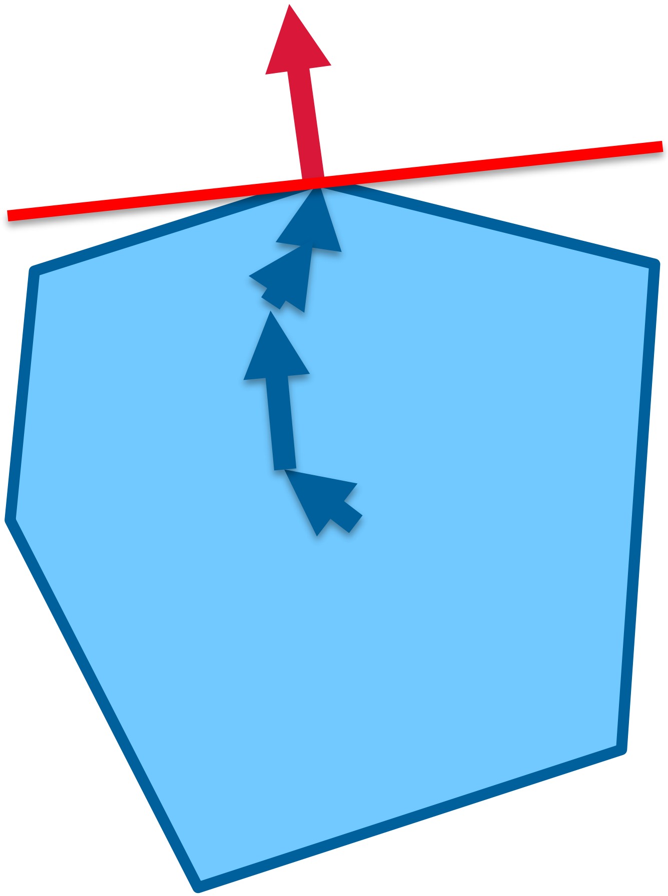
1. Start with initial set J_0 containing a critical node.
2. Rank other nodes according to connectivity.
3. Add most strongly connected node and repeat.

Interior point algorithm

$$\min\ c^T x$$

$$\text{s.t.}\quad Ax = b$$

$$x \geq 0$$



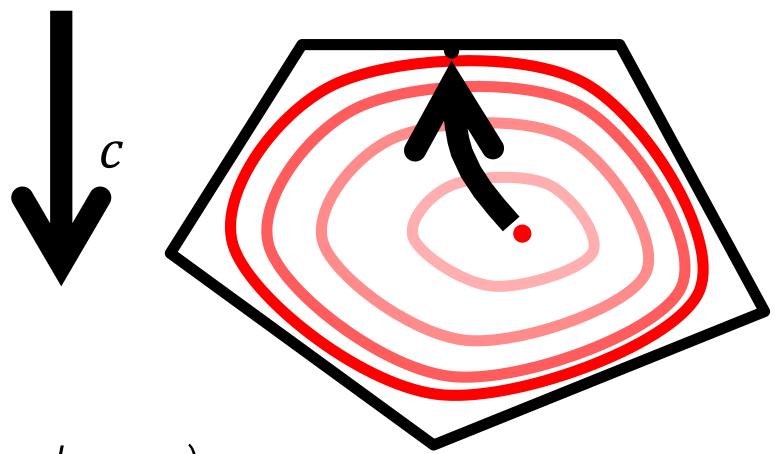
- Iterate traverses the interior of the feasible region
- It follows the **central path**
- ~ O(log n) iterations

The Barrier Algorithm and the Analytic Center

$$\min\ c^T x - \mu \sum_{j=1}^{n} \ln x_j$$ *(log barrier)*

$$\text{s.t.}\quad Ax = b$$

~~$x \geq 0$~~



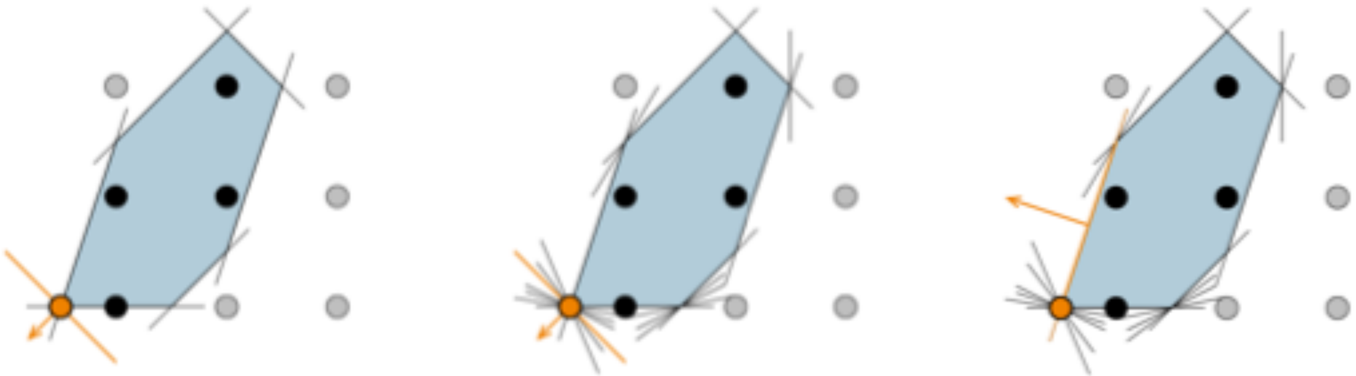
Solve KKT system:

$$\begin{aligned} Ax &= b \\ A^T y - s &= c \\ x_j s_j &= \mu \quad j = 1, \ldots, n \\ x, s &\geq 0 \end{aligned}$$

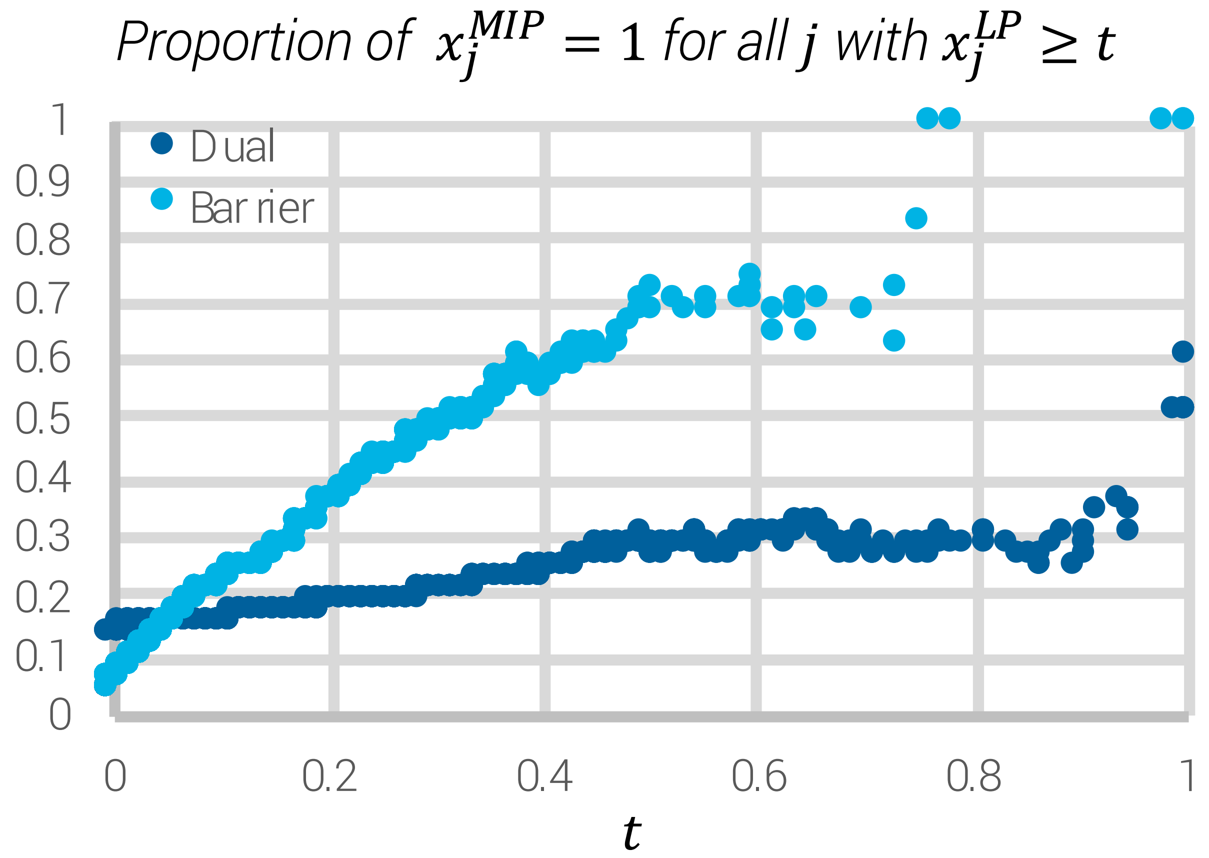
(complementary slackness)

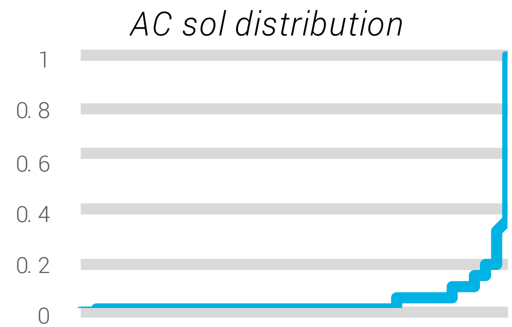
Analytic Center

- Strong convexity of log implies uniqueness
- Maximizes distance to boundary
- Can be computed by Barrier algorithm
- Without objective: Analytic center of polytope
- With objective: Analytic center of optimal face.
  - Interesting for highly dual degenerate problems.



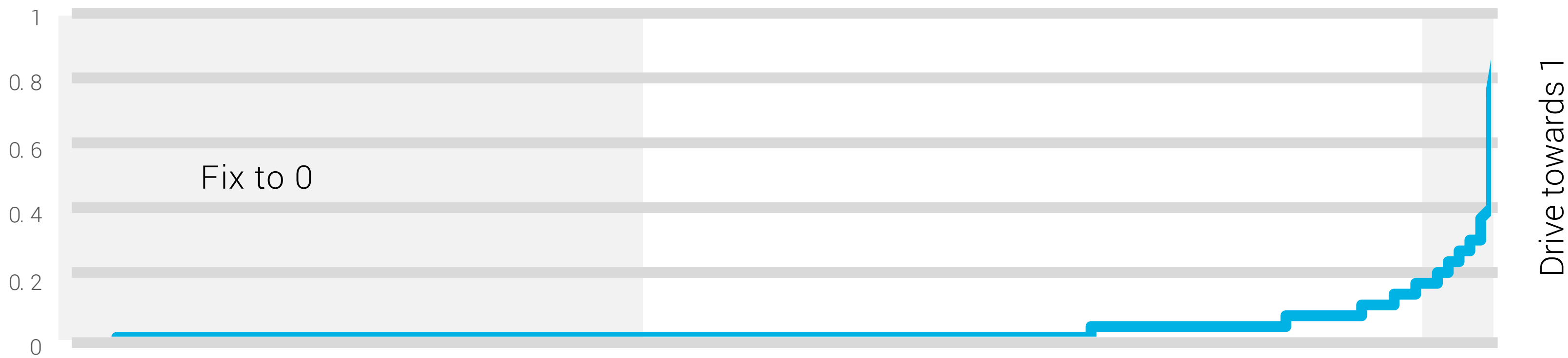
Analytic Center Heuristic (example)





Analytic Center Heuristic

- "Classic", general integer interpretation of Analytic Center:
  - Middle of polyhedron likely to have feasible solutions in its vicinity ➔ try rounding from the analytic center
- Our (binary-focused) interpretation:
  - Indicates the direction into which a variable is likely to move towards feasibility
  - Particularly interesting for variables that are likely to be 1 in a binary problem
  - Still, often not all of them can be set to 1 simultaneously



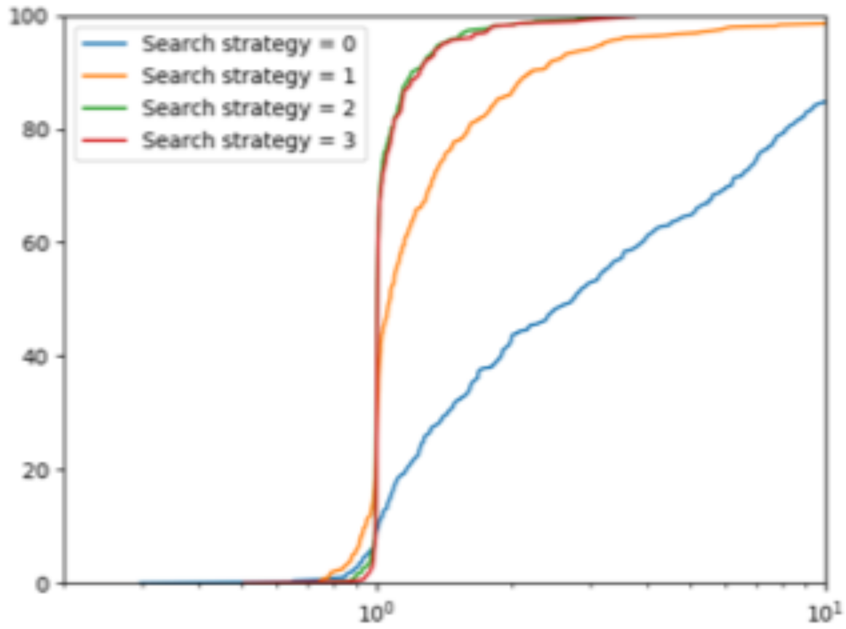
Analytic Center Heuristic

- Apply “soft rounding” by using the analytic center solution as auxiliary objective
  - Set objective coefficients proportionally to analytic center solution values
  - Tentatively fix some variables that are very close to one of their bounds
  - Apply restricted MIP solve

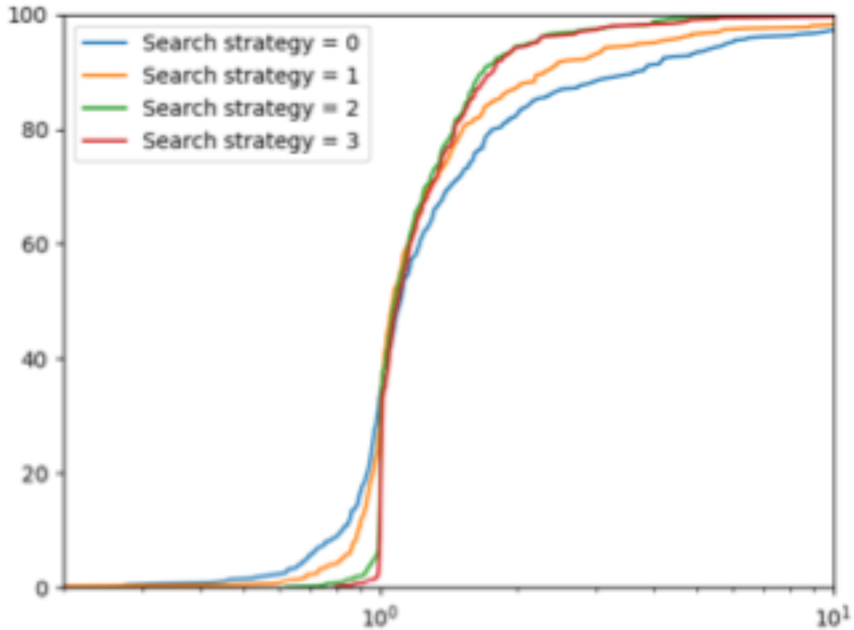
- Disregard original objective when creating analytic center solution
  - Makes heuristic useful for finding first feasible solution
  - Can be expensive, cf. Local Branching
  - Nicer interpretation than zero objective or objective flipping

42% benefit on H. Mittelmann’s Feasibility Benchmark

Performance profiles

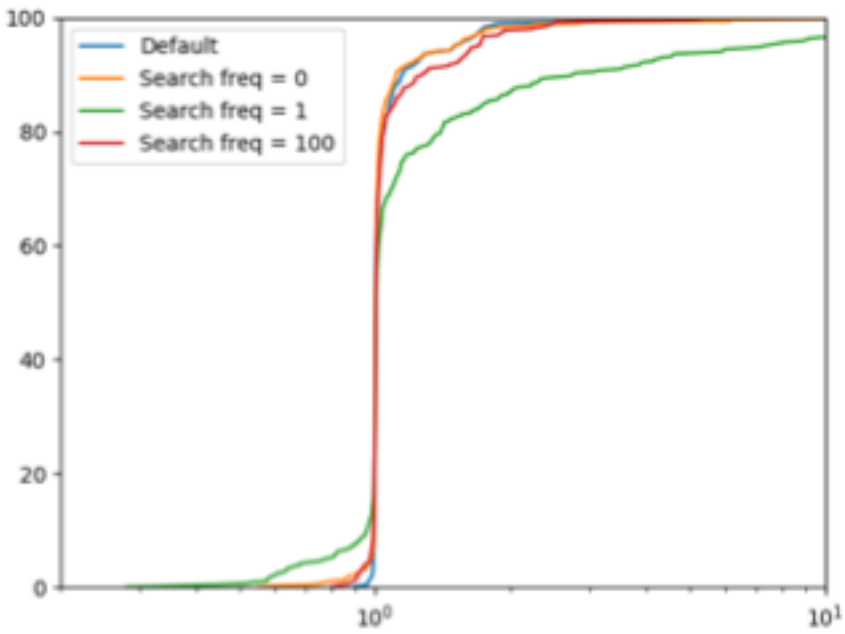


Primal dual integral

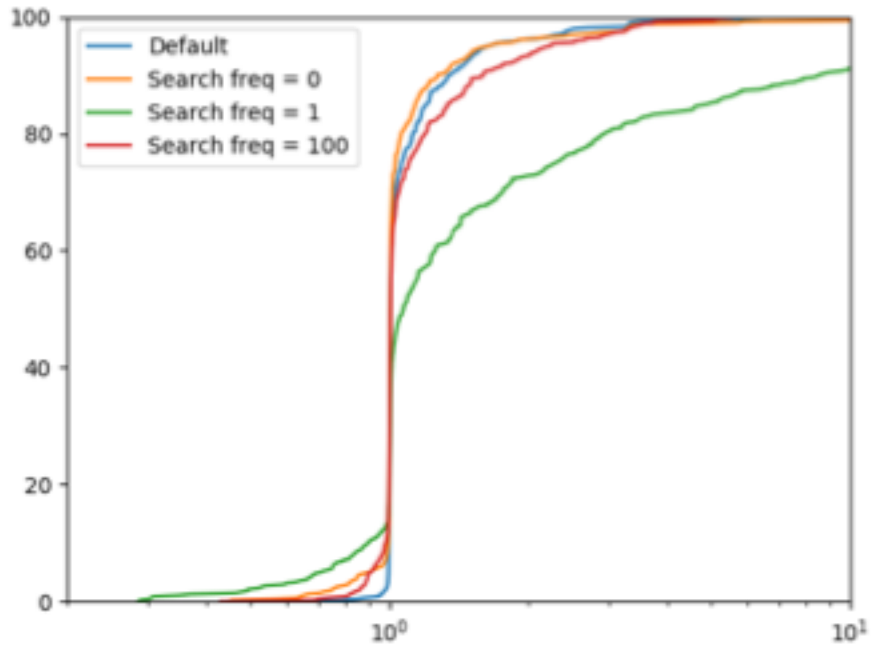


CPU Time

Performance profiles



Primal dual integral



CPU Time

Thank You

Pietro Belotti
pietrobelotti@fico.com