Solving the *seymour* problem from MIPLIB

Gábor Pataki

Dept. of OR
UNC, Chapel Hill

Stefan Schmieta, *Dept. of IE/OR, Columbia University*
Sebastián Ceria, *Columbia Business School and Dash Optimization*
Michael Ferris, *University of Wisconsin*
Jeff Linderoth, *Argonne National Laboratory*

The *seymour* problem

- Hard set-covering problem with 4944 rows, 1372 variables.
- Purpose: Finding minimal set of *irreducible configurations* in the proof of the 4-colour theorem.
- Absolute integrality gap is ≤ 19.16 (LP: 403.84; best IP solution: 423.00).
- A good case study in solving hard IP's.

Tools used to solve it

1. Branch-and-bound.

2. Cutting with disjunctive cuts.

3. Preprocessing, and decomposition.

Using branch-and-bound

- 1996: CPLEX 4.0 for $\approx$ 1000 wall-clock hours. Gap closed: **8.92** (G. Astfalk, HP).

A better choice is to use

- *Strong branching:* (ABCC, CPLEX) Compute penalties for 10 candidate variables, by doing 50 dual simplex pivots on both branches. Pick the variable with the best penalty. Gap closed by CPLEX 5.0 within 100,000 nodes: $\approx$ 9.

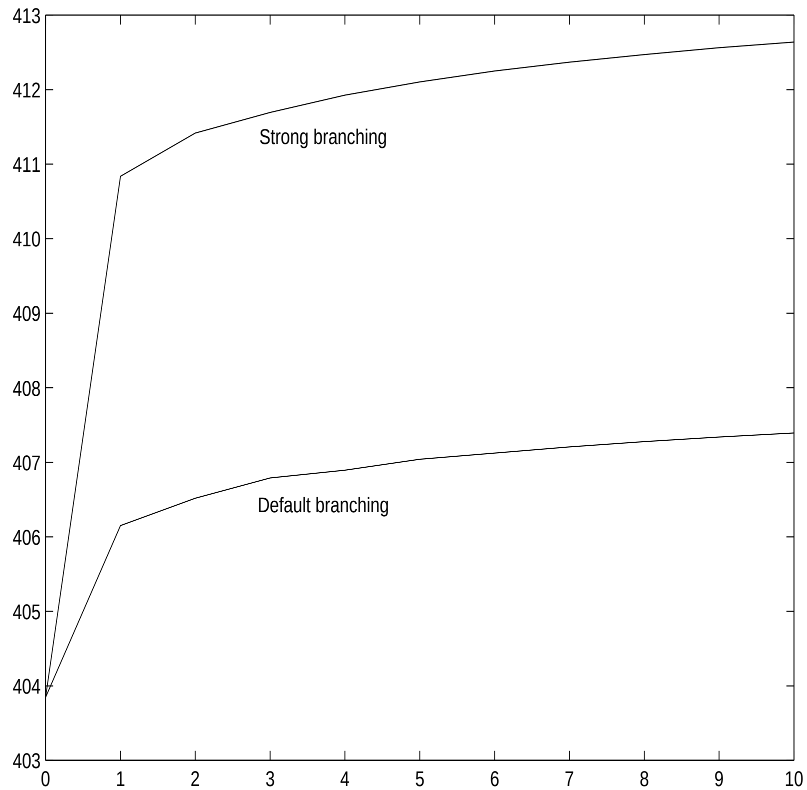

Figure 1: Default vs. strong branching on the *seymour* problem

Using disjunctive cuts

Given $\bar{x}$, an optimal solution to

$$\begin{aligned} Min \quad & cx \\ st. \quad & Ax \geq e \end{aligned}$$

- B&B : picks a variable to branch.
- L&P : picks ≈ 100 variables to generate $\alpha^i x \geq \beta_i$
  - valid for conv $(\{\, Ax \geq e,\, x_i = 0\,\} \cup \{\, Ax \geq e,\, x_i = 1\,\})$
  - violated by $\bar{x}$.

from the set of fractional variables.

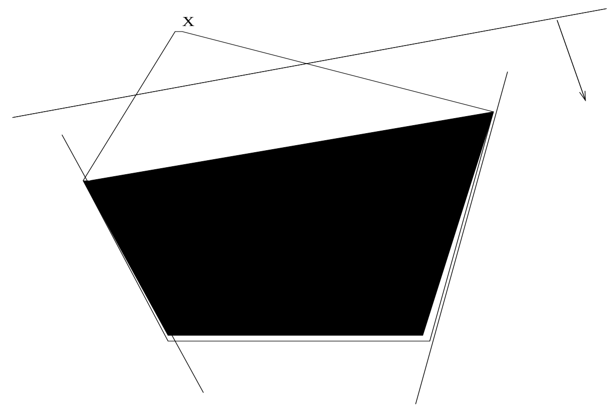

Figure 2: A disjunctive cut

Using disjunctive cuts

Fact: SB works $\Leftrightarrow$ disjunctive cuts work. Reason: SB picks the *best* of a set of disjunctions. Disjunctive cutting applies *many* of them. Both try to improve the effect of branching.

- Disjunctive cuts for 10 rounds, 100 cuts in each round (6 hours). Gap closed: **9.45**.

How to select the best cutting variables?

- We have $\approx$ 600 variables to generate cuts from. Which 100 are the best?

Options:

(1) Select the most fractional ones (closest to 0.5). Gap closed: **9.45**.

(2) Select them by computing SB penalties. Test 200 variables with 50 dual simplex pivots, pick the 100 with the best penalties. Gap closed: **10.28**.

(3) Same as (2), but test 400 variables with 100 pivots ...

Fact:

Time for testing variables $<$ Time for generating cuts $<<$

Time incurred by creating harder LP's by adding cuts.

It may be worth

- generating all 600, then
- selecting the best *afterwards*.

How to select the best cuts?

- We have ≈ 600 cuts, all violated by the current solution $\bar{x}$. Which 100 are the best?

Options:

(1) Select the 100 most violated, also prefer sparser ones, etc.

(2) Select the 100 with the best euclidean distance.

$$\text{dist}\ (\bar{x}, \{\, x \mid a^T x = \beta \,\}) = \frac{\beta - a^T \bar{x}}{\| a \|}$$

(3) Select the 100 with the best dual steepest edge prices.

(4) Select the 100 by usage within dual simplex with steepest edge pricing. If a cut is pivoted on, mark it. Continue, until

- 100 cuts have been marked, or
- the problem has been fully reoptimized.

Conclusion:

(1) Out of 600 cuts, less than 300 are ever pivoted on !

(2) Selecting 250 by usage works the best. Additional advantage: sparser cuts get selected this way.

(3) Gap closed by 10 rounds: ≈ 12.5.

Which point to cut off?

- Gap closed with same cut selection strategy, but cutting off an interior point: ≈ 13.0.

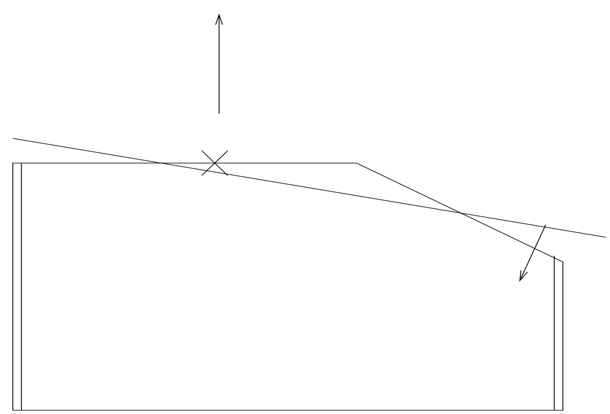

Figure 3: Cutting off an interior point optimal solution

Cuts + CPLEX on the strengthened formulation raises the gap by 16 units total.

Raising the bound by 16 units with cutting and some branching would do the job!

Preprocessing

Deleting dominated rows and columns reduces the problem from

- from $(m = 4944, m = 1372, nnz = 33549)$ to $(m = 4323, n = 882, nnz = 27987)$.
- The preprocessed problem is equally hard for cutting, and branch-and-bound.
- But: Preprocessing works well within branch-and-bound (Forrest, Ladanyi). Gap closed,
  - By our branch-and-bound after 50,000 nodes, using SB, no cuts: ≈ 8.
  - BY CPLEX after 50,000 nodes, using SB: ≈ 9; difference may be due to CPLEX generating clique cuts.

Decomposition

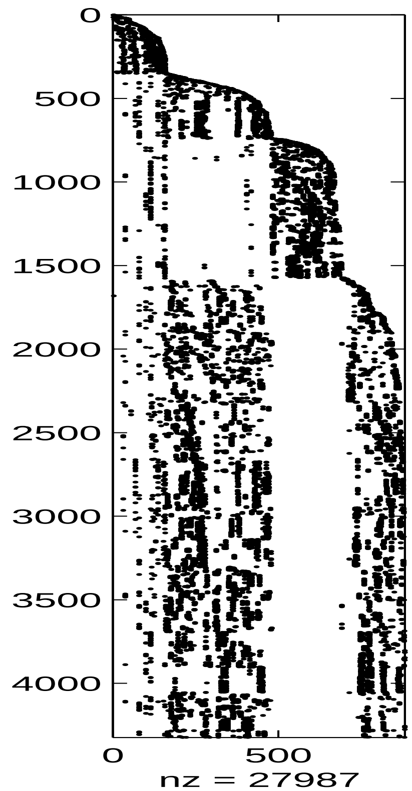

Figure 4: The matrix after preprocessing

:-) The matrix decomposes into 2 independent blocks!

:-(But the smaller one has only 18 variables . . .

:-) But it has a 1 unit gap, and solves in a minute!

:-) $\Rightarrow$ with no work, we reduced the gap to be closed by 1!

- :-(Not quite . . . since
- If cutting/branching closes the gap by x units on the original problem, it closes the gap by $x - 1$ units on the reduced problem . . .
- Reason: we have already solved the smaller problem without noticing it!

The final run

Goal: generate a small number of nodes within branch-and-cut, with $\geq$ units of gap closed.

- 10 rounds of cutting + 4 levels of branching + 2 rounds of cutting + 4 levels of branching + 1 round of cutting.
- We used preprocessing throughout only on the setcovering constraints.
- We generated 256 nodes, best: 16.77; worst: 15.17; median: 16.29.

The 256 nodes were solved on the *Condor* computing platform at Wisconsin and Argonne. Total wall clock time, including generating the 256 nodes, and solving them: $\approx$ 8000 hours.

Conclusion: The optimal solution is indeed 423.

Remarks

- Seymour's solution of value 423 was found only very late; most nodes were run using the cutoff value 424.
- More reduction in time is still possible: mostly by generating better balanced nodes.