Mixed-Integer Programming: It works better than you may think

Robert E. Bixby

Gurobi Optimization

A Definition

A *mixed-integer program* (MIP) is an optimization problem of the form

$$\begin{array}{ll} \textit{Minimize} & c^T x \\ \textit{Subject to} & Ax = b \\ & l \le x \le u \\ \multicolumn{2}{l}{\text{some or all } x_j \text{ integer}} \end{array}$$

Two Examples

Example 1: A Unit-Commitment Story

Electrical Power Industry, ERPI GS-6401, June 1989:
Mixed-integer programming (MIP) is a powerful modeling tool, “They are, however, theoretically complicated and computationally cumbersome”

In Other Words: MIP is an interesting “toy”, but it just isn’t going to work it practice.

An Example Unit-Commitment Model
California 7-Day Model

UNITCAL_7: 48939 constraints, 25755 variables (2856 binary)

Reported Results 1999 – machine unknown
2 Day model: 8 hours, no progress
7 Day model: 1 hour to solve initial LP

Desktop PC -- ran full 7-day model
CPLEX 6.5 (1999): 22 minutes, optimal

Gurobi Optimization

California 7-Day Model

```
Gurobi Optimizer version 3.0.0

Read MPS format model from file unitcal_7.mps.bz2
Optimize a model with 48939 Rows, 25755 Columns and 127595 NonZeros
Presolved: 38804 Rows, 19960 Columns, 105627 Nonzeros

Root relaxation: objective 1.945018e+07, 18340 iterations, 0.60 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 1.9450e+07    0  721          - 1.9450e+07     -      -    2s
     0     0 1.9596e+07    0  559          - 1.9596e+07     -      -   16s
     0     0 1.9598e+07    0  487          - 1.9598e+07     -      -   20s
H    0     0                    2.066856e+07 1.9598e+07  5.18%     -   25s
    13    11 1.9669e+07    6  217 2.0669e+07 1.9602e+07  5.16%   649   30s
    36    28 1.9668e+07    9  219 2.0669e+07 1.9605e+07  5.15%   707   35s
H   93    84                    1.998399e+07 1.9605e+07  1.90%   342   37s
   100    74 1.9678e+07   17  204 1.9984e+07 1.9606e+07  1.89%   321   43s
*  271   111              54    1.972042e+07 1.9606e+07  0.58%   170   43s
H  417    29                    1.964604e+07 1.9606e+07  0.21%   129   43s
   858   178 1.9629e+07   11  141 1.9637e+07 1.9609e+07  0.14%  96.9   50s
H  924   187                    1.963578e+07 1.9609e+07  0.13%  94.6   51s
H  987   221                    1.963563e+07 1.9611e+07  0.12%  93.9   53s
  1024   237 1.9630e+07   19  107 1.9636e+07 1.9611e+07  0.12%  93.5   56s
H 1034   237                    1.963556e+07 1.9611e+07  0.12%  92.8   56s
  1144   288 1.9630e+07   14  501 1.9636e+07 1.9611e+07  0.12%  89.0   63s
  1147   290 1.9617e+07   13  595 1.9636e+07 1.9611e+07  0.12%  88.8   71s
  1153   294 1.9626e+07   17  491 1.9636e+07 1.9611e+07  0.12%  88.3   80s
  1163   303 1.9611e+07   16  547 1.9636e+07 1.9611e+07  0.12%   120  156s
  1170   303 1.9624e+07   20  488 1.9636e+07 1.9611e+07  0.12%   123  166s
  1245   294 1.9621e+07   30  399 1.9636e+07 1.9619e+07  0.09%   131  185s
  1673   261 1.9623e+07   35  120 1.9636e+07 1.9623e+07  0.07%   117  190s

Cutting planes:
  Gomory: 20
  Cover: 31
  Implied bound: 553
  Clique: 61
  MIR: 71
  Flow cover: 416

Explored 2167 nodes (274011 simplex iterations) in 194.37 seconds
Optimal solution found (tolerance 1.00e-04)
```

Gurobi Optimization

Example 2: A supply-chain model

- Model description:
  - Weekly model, daily buckets: Objective to minimize end-of-day inventory.
  - Production (single facility), inventory, shipping (trucks), wholesalers (demand known)
- Initial modeling phase
  - Simplified prototype + complicating constraints (production run grouping req't, min truck constraints)
  - RESULT: Couldn't get good feasible solutions.
- Decomposition approach
  - Talk to current scheduling team: They first decide on "producibles" schedule. Simulate using heuristics.
  - Fixed model: Fix variables and run MIP

Supply-chain scheduling (continued): Solving the fixed model

CPLEX 5.0 (1997):

```
Integer optimal solution (0.0001/0):  Objective =    1.5091900536e+05
Current MIP best bound =    1.5090391809e+05 (gap = 15.0873)
Solution time = 3465.73 sec.  Iterations = 7885711  Nodes = 489870 (2268)
```

Gurobi 2.0 (2009):

```
Cutting planes:
  Gomory: 16
  Implied bound: 33
  MIR: 1
  Flow cover: 70

Explored 9 nodes (4517 simplex iterations) in 0.49 seconds

Optimal solution found (tolerance 1.00e-04)
Best objective 1.5091900536e+05, best bound 1.5090900608e+05, gap 0.0066%
```

Original model: Now solves to optimality in 40 minutes (20% improvement in solution quality)

What Happened?

(in 1999)

Computational History: 1950 –1998

- **1954 Dantzig, Fulkerson, S. Johnson:** 42 city TSP
  - Solved to optimality using LP and cutting planes
- **1957 Gomory**
  - Cutting plane algorithms
- **1960 Land, Doig, 1965 Dakin**
  - B&B
- **1971 MPSX/370**
- **1972 UMPIRE**
  - LP-based B&B
  - MIP became commercially viable

- 1972 – 1998 Good B&B remained the state-of-the-art in commercial codes, in spite of ….
  - Edmonds, polyhedral combinatorics
  - 1973 Padberg, cutting planes
  - 1973 Chvátal, revisited Gomory
  - 1974 Balas, disjunctive programming
  - 1983 Crowder, Johnson, Padberg: PIPX, pure 0/1 MIP
  - 1987 Van Roy and Wolsey: MPSARX, mixed 0/1 MIP
  - TSP, Grötschel, Padberg, …

1998 … A New Generation of MIP Codes

- Linear programming
  - Stable, robust dual simplex
- Variable/node selection
  - Influenced by traveling salesman problem
- Primal heuristics
  - 12 different tried at root
  - Retried based upon success
- Node presolve
  - Fast, incremental bound strengthening (very similar to Constraint Programming)
- Presolve – numerous small ideas
  - Probing in constraints:

    $\sum x_j \le (\sum u_j)\, y, \quad y = 0/1$

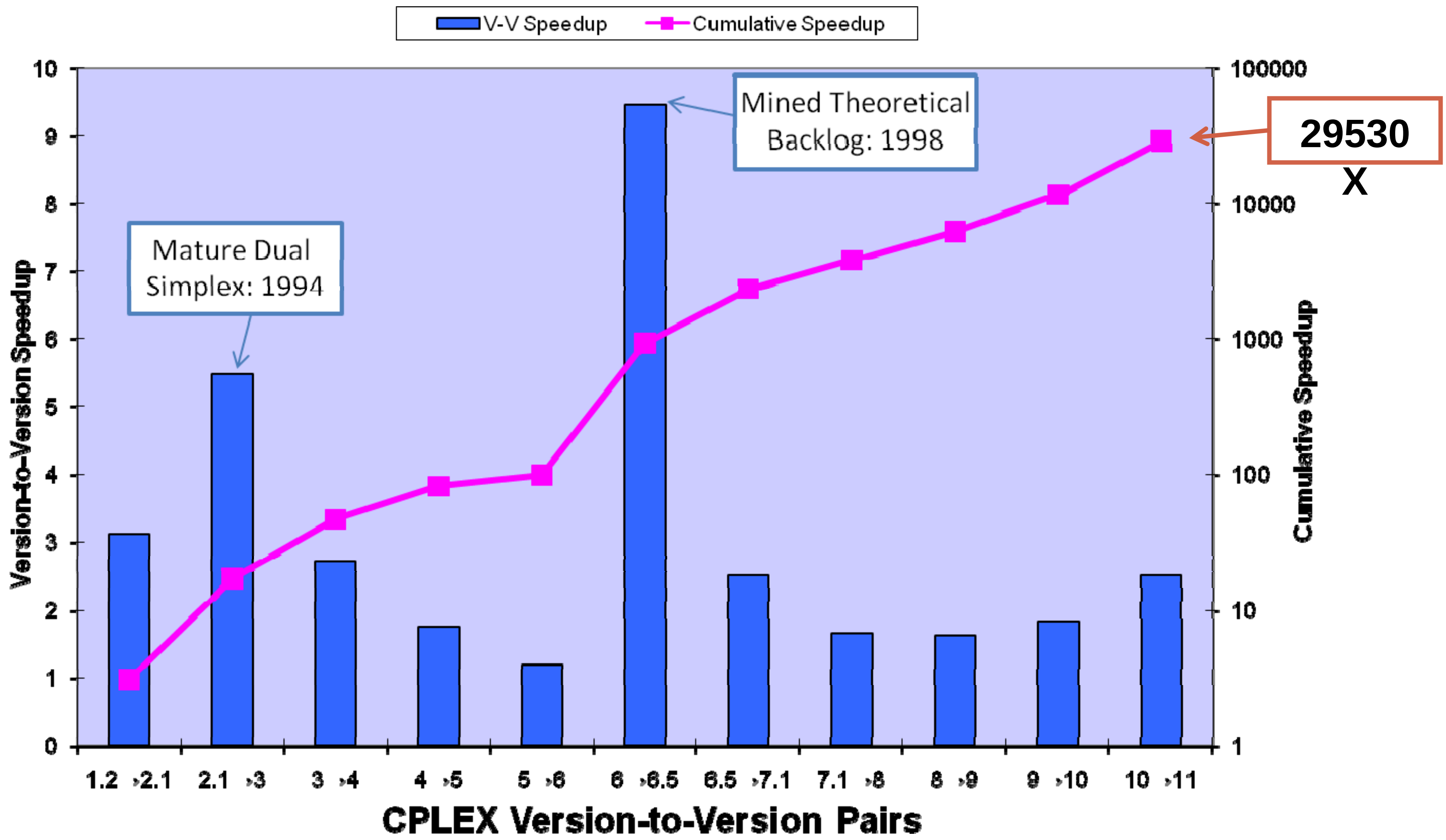
    $\Rightarrow x_j \le u_j y$ (for all j)
- Cutting planes
  - Gomory, mixed-integer rounding (MIR), knapsack covers, flow covers, cliques, GUB covers, implied bounds, zero-half cuts, path cuts

Some Test Results

- Test set: 1852 real-world MIPs
  - Full library
    - 2791 MIPs
  - Removed:
    - 559 "Easy" MIPs
    - 348 "Duplicates"
    - 22 "Hard" LPs (0.8%)
- Parameter settings
  - Pure defaults
  - 30000 second time limit
- Versions Run
  - CPLEX 1.2 (1991) -- CPLEX 11.0 (2007)

Gurobi Optimization

MIP Performance Improvements
1991-2007



What Has Happened Since 2007?

Gurobi Optimization

- Gurobi Optimization, Inc.
  - Incorporated July, 2008
  - Founders: Zonghao Gu, Ed Rothberg, Bob Bixby
- Product Releases
  - Version 1.0: May 2009
    - LP simplex & MIP
    - **Public benchmarks: ~ CPLEX 11.0**
  - Version 2.0: October 2009
  - Version 3.0: April 2010

MIP Performance – Gurobi

- Gurobi V1.0 → V2.0
  - 2340 total models in test set
    - 1309 solved by both in < 1 second (removed)
    - 650 solved by at least one in < 10000 seconds
    - 381 solved by neither in < 10000 seconds
- Gurobi V2.0 → V3.0
  - 2458 total models in test set
    - 1350 solved by both in < 1 second (removed)
    - 794 solved by at least one in < 10000 seconds
    - 314 solved by neither in < 10000 seconds

MIP Performance – Gurobi

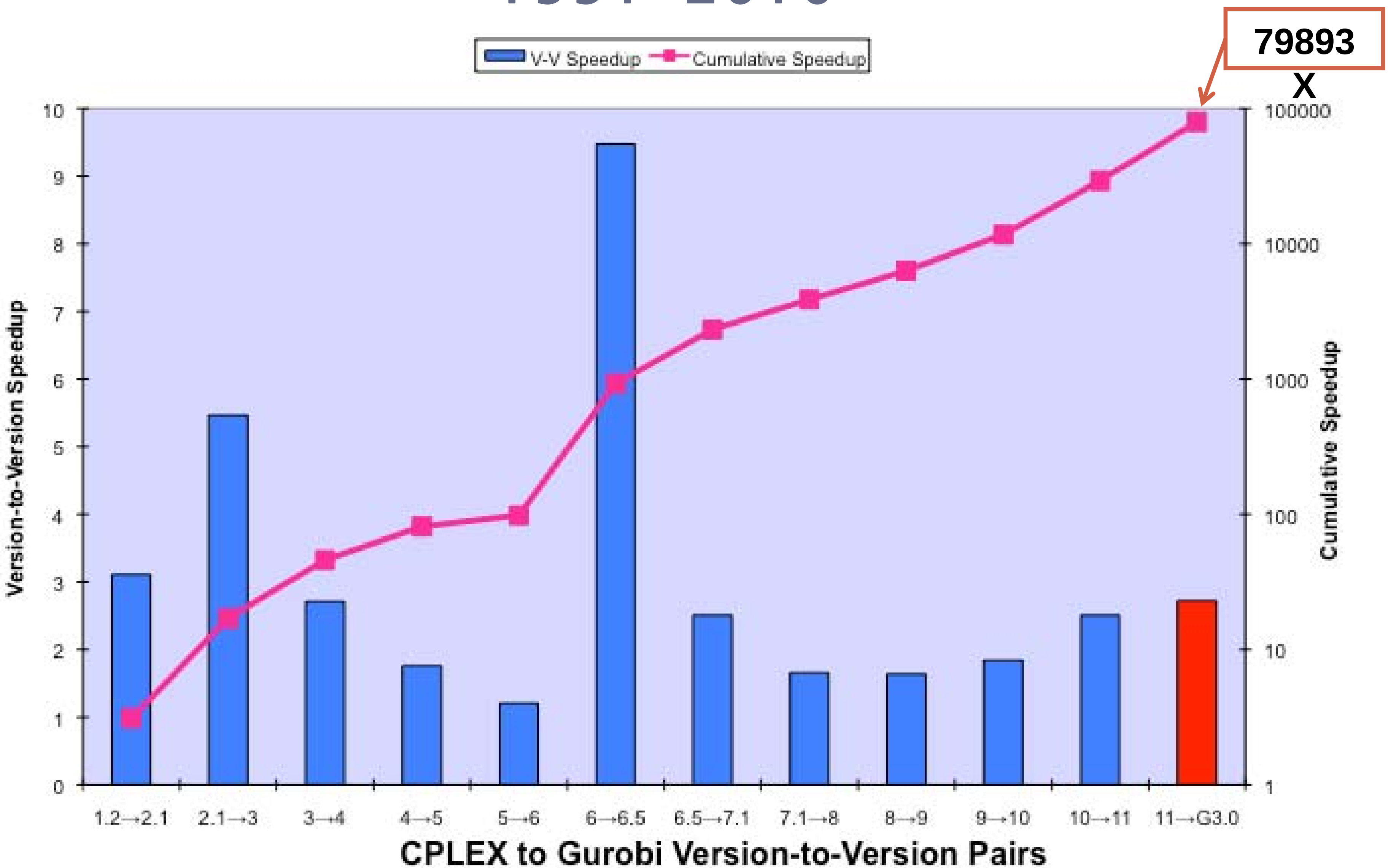
- Gurobi V1.0 → V2.0

Time	# Models	Speedup
> 1s	650	1.7x
> 10s	410	1.9x
> 100s	210	2.2x
> 1000s	59	3.9x

- Gurobi V2.0 → V3.0

Time	# Models	Speedup
> 1s	794	1.6x
> 10s	521	2.0x
> 100s	295	2.9x
> 1000s	144	6.7x

MIP Performance Improvements
1991–2010



Publicly Available Table http://scip.zib.de/