Mixed Integer Programming:

Analyzing 12 Years of Progress

Roland Wunderling
CPLEX Optimizer Architect
Background

- 2001: Manfred Padberg’s 60th birthday
    - Analysis of the relative contributions of the key ingredients of Branch-and-Cut Algorithms for solving MIPs

- 2013: Martin Grötschel’s 65th birthday
    - What stayed?
    - What changed?
    - Why?
Agenda

- Methodology
  - How to Benchmark
  - How to Measure Importance of Features

- Analysis
  - Presolving
  - Cuts
  - Heuristics
  - Parallelism

- Summary
Benchmarking

- Run competing algorithms on set of problem instances
  - Measure and compare runtime, timeouts
  - Use geometric mean for aggregation

- Performance Variability
  - Seemingly performance neutral changes (random seed, platform, permutation of variables, ...) have drastic impact on solution time
  - Has been observed for a long time, e.g.

  - Emilie Danna: Performance variability in mixed integer programming, Presentation at Workshop on Mixed Integer Programming 2008
    - Friend or foe for Solvers?
  - Tuesday Oct 08, 08:00 - 09:30, Andrea Lodi: Performance Variability in Mixed-integer Programming
  - Wednesday Oct 09, 15:30 - 17:00, Andrea Tramontani: Concurrent root cut loops to exploit random performance variability
    - Definitely foe for Benchmarking
A benchmarking myth

- Compare Solver S against reference Solver R
  - Solver S claimed to be faster than R on “hard problems”

- Model Set M
  - Solution times $t_R(m)$, $m \in M$, for S and R are in (0sec, 100sec]

- Classify Models in to hard models H and easy models E
  - $m \in H$, if $t_R(m) > 80$sec
  - $m \in E$, otherwise

- Computational confirmation of speedup:
  - 1.8x faster on hard models
  - 0.8x slower on easy models
A benchmarking myth

- Compare Solver S against reference Solver R
  - Solver S claimed to be faster than R on “hard problems”

- Model Set M
  - Solution times $t_R(m)$, $m \in M$, for S and R are in (0sec, 100sec]

- Classify Models into hard models H and easy models E
  - $m \in H$, if $t_R(m) > 80$sec
  - $m \in E$, otherwise

- Computational confirmation of speedup:
  - 1.8x faster on hard models
  - 0.8x slower on easy models

- Times for S and R uniform random numbers:
  - $<t_R(E)> = 40$  $<t_R(H)> = 90$
  - $<t_S(E)> = 50$  $<t_S(H)> = 50$
  - Speedup: 4/5  9/5
A benchmarking myth

- Compare Solver $S$ against reference Solver $R$
  - Solver $S$ claimed to be faster than $R$ on “hard problems”

- Model Set $M$
  - Solution times $t_R(m)$, $m \in M$, for $S$ and $R$ are in $(0\text{sec}, 100\text{sec}]$

- Classify Models $M$ to hard models $H$ and easy models $E$
  - $m \in H$, if $t_R(m) > 80\text{sec}$
  - $m \in E$, otherwise

- Computational confirmation of speedup:
  - 1.8x faster on hard models
  - 0.8x slower on easy models

- Times for $S$ and $R$ uniform random numbers:
  - $<t_R(E)> = 40$ $<t_R(H)> = 90$
  - $<t_S(E)> = 50$ $<t_S(H)> = 50$
  - Speedup: 4/5 9/5
Avoiding the bias

- The problem is real: 2 different random seeds for CPLEX 12.5
- Problem comes from using times from one solver to define subsets of problems

<table>
<thead>
<tr>
<th>biased subsets</th>
<th># of problems</th>
<th>geomean</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>3159</td>
<td>1.00</td>
</tr>
<tr>
<td>[0,10k]</td>
<td>3082</td>
<td>1.00</td>
</tr>
<tr>
<td>[1,10k]</td>
<td>1848</td>
<td>0.99</td>
</tr>
<tr>
<td>[10,10k]</td>
<td>1074</td>
<td>0.95</td>
</tr>
<tr>
<td>[100,10k]</td>
<td>552</td>
<td>0.87</td>
</tr>
<tr>
<td>[1k,10k]</td>
<td>207</td>
<td>0.76</td>
</tr>
</tbody>
</table>
Avoiding the bias

- The problem is real: 2 different random seeds for CPLEX 12.5
- Problem comes from using times from one solver to define subsets of problems
- Solution
  - Use (max) times from all solvers to define subsets of problems

<table>
<thead>
<tr>
<th>biased subsets</th>
<th># of problems</th>
<th>geomean</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>3159</td>
<td>1.00</td>
</tr>
<tr>
<td>[0,10k]</td>
<td>3082</td>
<td>1.00</td>
</tr>
<tr>
<td>[1,10k]</td>
<td>1848</td>
<td>0.99</td>
</tr>
<tr>
<td>[10,10k]</td>
<td>1074</td>
<td>0.95</td>
</tr>
<tr>
<td>[100,10k]</td>
<td>552</td>
<td>0.87</td>
</tr>
<tr>
<td>[1k,10k]</td>
<td>207</td>
<td>0.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>unbiased subsets</th>
<th># of problems</th>
<th>geomean</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>3159</td>
<td>1.00</td>
</tr>
<tr>
<td>[0,10k]</td>
<td>3082</td>
<td>1.00</td>
</tr>
<tr>
<td>[1,10k]</td>
<td>1879</td>
<td>1.00</td>
</tr>
<tr>
<td>[10,10k]</td>
<td>1121</td>
<td>1.01</td>
</tr>
<tr>
<td>[100,10k]</td>
<td>604</td>
<td>1.01</td>
</tr>
<tr>
<td>[1k,10k]</td>
<td>238</td>
<td>1.08</td>
</tr>
</tbody>
</table>
Avoiding the bias

- The problem is real: 2 different random seeds for CPLEX 12.5
- Problem comes from using times from one solver to define subsets of problems
- Solution
  - Use (max) times from all solvers to define subsets of problems
- Note
  - 250 models can not measure performance difference of less than 10%
  - Will use [10,10k] bracket

<table>
<thead>
<tr>
<th>biased subsets</th>
<th># of problems</th>
<th>geomean</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>3159</td>
<td>1.00</td>
</tr>
<tr>
<td>[0,10k]</td>
<td>3082</td>
<td>1.00</td>
</tr>
<tr>
<td>[1,10k]</td>
<td>1848</td>
<td>0.99</td>
</tr>
<tr>
<td>[10,10k]</td>
<td>1074</td>
<td>0.95</td>
</tr>
<tr>
<td>[100,10k]</td>
<td>552</td>
<td>0.87</td>
</tr>
<tr>
<td>[1k,10k]</td>
<td>207</td>
<td>0.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>unbiased subsets</th>
<th># of problems</th>
<th>geomean</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>3159</td>
<td>1.00</td>
</tr>
<tr>
<td>[0,10k]</td>
<td>3082</td>
<td>1.00</td>
</tr>
<tr>
<td>[1,10k]</td>
<td>1879</td>
<td>1.00</td>
</tr>
<tr>
<td>[10,10k]</td>
<td>1121</td>
<td>1.01</td>
</tr>
<tr>
<td>[100,10k]</td>
<td>604</td>
<td>1.01</td>
</tr>
<tr>
<td>[1k,10k]</td>
<td>238</td>
<td>1.08</td>
</tr>
</tbody>
</table>
Measuring Impact

- MIP is a bag of tricks
  - Presolving
  - Cutting planes
  - Branching
  - Heuristics
  - ...

- How important is each trick?
  Compare runs with feature turned on and off
  - Solution time degradation (geometric mean)
  - # of solved models
    • Essential or just speedup?
  - Number of affected models
    • General or problem specific?

<table>
<thead>
<tr>
<th>Feature</th>
<th>Degradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>No cuts</td>
<td>53.7x</td>
</tr>
<tr>
<td>No presolve</td>
<td>10.8x</td>
</tr>
<tr>
<td>Trivial branching</td>
<td>2.9x</td>
</tr>
<tr>
<td>No heuristics</td>
<td>1.4x</td>
</tr>
</tbody>
</table>
Component Impact CPLEX 12.5 Summary

Benchmarking setup
• 1769 models
• 12 core Intel Xenon 2.66 GHz
• Unbiased: At least one of all the test runs took at least 10 sec

% affected
- no presolve: 99%
- most inf branching: 82%
- no cuts: 91%
- no parallelism: 91%
- no dynamic search: 83%
- no heuristics: 93%
- no symmetry: 26%
- no conflict analysis: 46%
- no pumps: 65%
Component Impact CPLEX 12.5 Summary

**Fundamental Features**
- Lots of models unsolvable without
- Apply to most models
Component Impact CPLEX 12.5 Summary

Important Features

- Many models unsolvable without
- Apply to most models

- no presolve: 7.57%
- most inf branching: 19.3%
- no cuts: 37.5%
- no parallelism: 23.3%
- no dynamic search: 1.55
- no heuristics: 1.51
- no symmetry: 1.15
- no conflict analysis: 1.09
- no pumpspeed: 1.06

% affected

© 2013 IBM Corporation
Parallelism is *not* important

- turning off == 12x fewer cycles
  i.e. just a tighter time limit
- Hardware cannot defeat combinatorial explosion

Component Impact CPLEX 12.5 Summary
Component Impact CPLEX 12.5 Summary

Special Features
- Few models unsolvable without
- Apply to few models

Additional timeouts

% affected: 99% 82% 91% 91% 83% 93% 26% 46% 65%

- no presolve
- most int branching
- no cuts
- no parallelism
- no dynamic search
- no heuristics
- no symmetry
- no conflict analysis
- no pumpreduce
Component Impact CPLEX 12.5 Summary

- Only 106 models
- Biased towards Cuts due to selection of “hard” models for CPLEX 5.0
- Impact of other components matches

<table>
<thead>
<tr>
<th>Component</th>
<th>% affected</th>
<th>time ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>no presolve</td>
<td>99%</td>
<td>8.0</td>
</tr>
<tr>
<td>most inf branching</td>
<td>82%</td>
<td>5.1</td>
</tr>
<tr>
<td>no cuts</td>
<td>91%</td>
<td>2.6</td>
</tr>
<tr>
<td>no parallelism</td>
<td>91%</td>
<td>1.0</td>
</tr>
<tr>
<td>no dynamic search</td>
<td>83%</td>
<td>1.0</td>
</tr>
<tr>
<td>no heuristics</td>
<td>93%</td>
<td>1.0</td>
</tr>
<tr>
<td>no symmetry</td>
<td>26%</td>
<td>1.0</td>
</tr>
<tr>
<td>no conflict analysis</td>
<td>46%</td>
<td>1.0</td>
</tr>
<tr>
<td>no pumpreduction</td>
<td>65%</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Component Impact CPLEX 12.5 – Presolve

additonal timeouts vs. time ratio

- no primal: 80.9
- no dual: 7.7
- no probing: 1.56
- no node presolve: 1.24
- no root restarts: 1.10
- no set cover probing: 1.04
- no variable goals: 1.04
- no component fixing: 1.03
- regular Tarjan SCC: 1.02
Component Impact CPLEX 12.5 Summary

- no presolve: 99%
- most inf branching: 82%
- no cuts: 91%
- no parallelism: 91%
- no dynamic search: 83%
- no heuristics: 93%
- no symmetry: 26%
- no conflict analysis: 46%
- no pmpreduce: 65%

Additional timeouts:
- no presolve: 470
- most inf branching: 470
- no cuts: 375
- no parallelism: 75
- no dynamic search: 75
- no heuristics: 75
- no symmetry: 75
- no conflict analysis: 75
- no pmpreduce: 75

Time ratio: 8
Component Impact CPLEX 12.5 – Cutting Planes

The chart shows the impact of various cutting plane components on the additional timeouts and time ratios. The cutting planes evaluated include:

- no MIR
- no Comorcy
- no knapsack covers
- no zero-half
- no implied bounds
- no coverson
- no cliques
- no flow paths
- no MCF
- no GUB covers
- cut-and-branch

The y-axis represents additional timeouts, while the x-axis shows different cutting plane components. The bars indicate the time ratio for each component.
Component Impact CPLEX 12.5 – Cutting Planes

Bixby et al. 2001

© 2013 IBM Corporation
Component Impact CPLEX 12.5 Summary

% affected

- no presolve: 99%
- root inf branching: 82%
- no cuts: 91%
- no parallelism: 91%
- no dynamic search: 83%
- no heuristics: 93%
- no symmetry: 26%
- no conflict analysis: 46%
- no pupred: 65%

Additional timeouts:

- no presolve: 450
- root inf branching: 440
- no cuts: 370
- no parallelism: 340
- no dynamic search: 270
- no heuristics: 150
- no symmetry: 130
- no conflict analysis: 120
- no pupred: 110

Time ratio:

- 4.5
- 3.5
- 3.0
- 2.5
- 2.0
- 1.5
- 1.0
- 0.5

© 2013 IBM Corporation
Component Impact CPLEX 12.5 – Primal Heuristics

- no starting: 1.48
- no before LP: 1.07
- no improvement: 1.28
- no RINS: 1.21
Component Impact CPLEX 12.5 Summary

<table>
<thead>
<tr>
<th>Component</th>
<th>% affected</th>
<th>Additional Timeouts</th>
</tr>
</thead>
<tbody>
<tr>
<td>no presolve</td>
<td>99%</td>
<td>450</td>
</tr>
<tr>
<td>most inf branching</td>
<td>82%</td>
<td>350</td>
</tr>
<tr>
<td>no cuts</td>
<td>91%</td>
<td>250</td>
</tr>
<tr>
<td>no parallelism</td>
<td>91%</td>
<td>150</td>
</tr>
<tr>
<td>no dynamic search</td>
<td>83%</td>
<td>70</td>
</tr>
<tr>
<td>no heuristics</td>
<td>93%</td>
<td>50</td>
</tr>
<tr>
<td>no symmetry</td>
<td>26%</td>
<td>30</td>
</tr>
<tr>
<td>no conflict analysis</td>
<td>46%</td>
<td>10</td>
</tr>
<tr>
<td>no pmpreduce</td>
<td>65%</td>
<td>5</td>
</tr>
</tbody>
</table>

Time ratio:

© 2013 IBM Corporation
Parallelism in CPLEX

- **Types of Parallelism**
  - Opportunistic Parallelism
  - Deterministic Parallelism
    - Identical runs produce same solution path and results
    - Use deterministic locks
    - Based on deterministic time
    - Implemented via counting memory accesses

- **Parallel Tasks**
  - Root node parallelism
    - Concurrent LP solve
    - Heuristics concurrent to cutting phase
    - Parallel Cut loop
  - Parallel Processing of B&C Tree
The Cost of Determinism
Date: 28 September 2013
Testset: 3147 models (1792 in ≥ 10 sec, 1554 in ≥ 100 sec, 1384 in ≥ 1000 sec)
Machine: Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads (deterministic since CPLEX 11.0)
Timelimit: 10,000 sec
Appendix
Component Impact CPLEX 12.5 Summary

- No presolve: 757 additional timeouts
- Most inf branching: 463 additional timeouts
- No cuts: 375 additional timeouts
- No parallelism: 233 additional timeouts
- No dynamic search: 155 additional timeouts
- No heuristics: 151 additional timeouts
- No symmetry: 115 additional timeouts
- No conflict analysis: 109 additional timeouts
- No pumpreduce: 106 additional timeouts
Component Impact CPLEX 12.5 – Branching

- No reliability: 1.05
- No pseudo-reduced costs: 1.06
- No non-linear

Additional timeouts

Time ratio