Analyzing the computational impact of individual MINLP solver components

Ambros M. Gleixner

joint work with Stefan Vigerske

Zuse Institute Berlin · Matheon · Berlin Mathematical School

MINLP 2014, June 4, Carnegie Mellon University

ZIB – Fast Algorithms, Fast Computers

Zuse Institute Berlin is a research institute and computing center of the State of Berlin with research units:

- Numerical Analysis and Modeling
- Visualization and Data Analysis
- Optimization: Energy–Traffic–Telecommunication–**Linear and Nonlinear IP**
- Scientific Information Systems
- Computer Science and High Performance Computing

What is SCIP?

SCIP (Solving Constraint Integer Programs) ...

- integrates
  - CP features (domain propagation)
  - MIP features (cutting planes, LP relaxation)
  - SAT-solving features (conflict analysis, restarts)
- is a branch-cut-and-price framework
- has an modular structure
- can be extended via plugins
- is free for academic purposes
- and is available in source-code under `http://scip.zib.de`
- provides a full-scale MIP and **MINLP solver**

Outline

Analyzing MINLP solver components

Benchmarking methodology

Separation

Reformulation

Primal Heuristics

Tree search

Propagation

Outline

Analyzing MINLP solver components

Benchmarking methodology

Separation

Reformulation

Primal Heuristics

Tree search

Propagation

789 choose 475

Collection of 789 publicly available MINLP instances

- ▷ MINLPLib2 α: MINLPLib+minlp.org+Bonmin+...

Hardware

- ▷ Dell PowerEdge M1000e, 48 GB RAM, Intel Xeon X5672@3.2 GHz

Software

- ▷ SCIP 3.1.0.1
- ▷ SoPlex 2.0
- ▷ Ipopt 3.11.8
- ▷ CppAD 20140000.1

475 test instances, 15 settings, 1 hour time limit

- ▷ 314 instances not solved by default within 2 hours

Averaging over heterogeneous test sets

Instances vary widely in size, nonlinearity, . . .

Averaging over heterogeneous test sets

Instances vary widely in size, nonlinearity, . . . , **time to optimality**

- arithmetic average: dominated by large times
- geometric average: weights trivial and hard instances equally
- shifted geometric average: which shift?

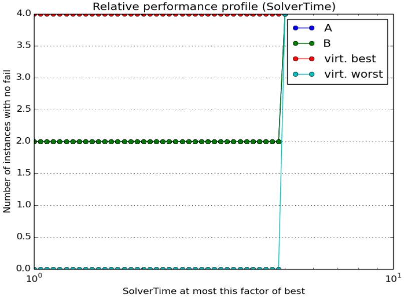
Averaging over heterogeneous test sets

Instances vary widely in size, nonlinearity, . . . , time to optimality

- arithmetic average: dominated by large times
- geometric average: weights trivial and hard instances equally
- shifted geometric average: which shift?

Some results are not distinguished by performance profiles alone:

inst	A	B
1	10s	2s
2	10s	2s
3	10s	50s
4	10s	50s



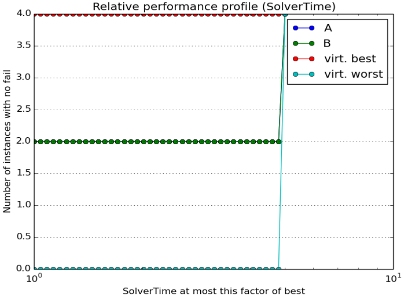
Averaging over heterogeneous test sets

Instances vary widely in size, nonlinearity, . . . , time to optimality

- arithmetic average: dominated by large times
- geometric average: weights trivial and hard instances equally
- shifted geometric average: which shift?

Some results are not distinguished by performance profiles alone:

inst	A	B
1	5x	1x
2	5x	1x
3	1x	5x
4	1x	5x



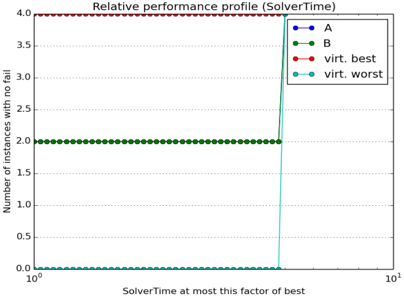
Averaging over heterogeneous test sets

Instances vary widely in size, nonlinearity, ..., time to optimality

- arithmetic average: dominated by large times
- geometric average: weights trivial and hard instances equally
- shifted geometric average: which shift?

Some results are not distinguished by performance profiles alone:

inst	A	B
1	10s	2s
2	20s	100s
3	50s	10s
4	100s	500s



Filtered Performance Diagrams

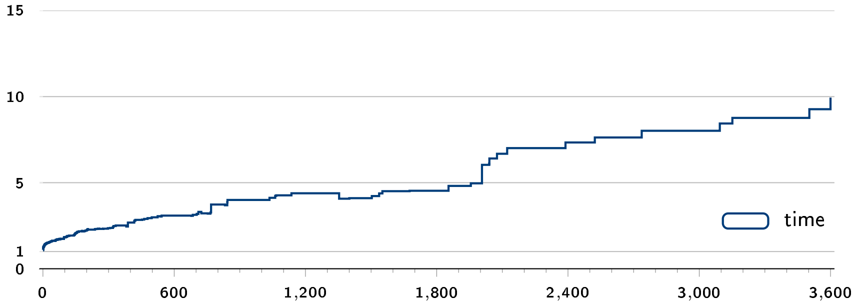
Gradually exclude instances **solved by A and B** and compute speedup:

$$t \mapsto \frac{\mu(\{t_{A,i} : \max\{t_{A,i}, t_{B,i}\} \geq t\})}{\mu(\{t_{B,i} : \max\{t_{A,i}, t_{B,i}\} \geq t\})}$$

Filtered Performance Diagrams

Gradually exclude instances **solved by A and B** and compute speedup:

$$t \mapsto \frac{\mu(\{t_{A,i} : \max\{t_{A,i}, t_{B,i}\} \geq t\})}{\mu(\{t_{B,i} : \max\{t_{A,i}, t_{B,i}\} \geq t\})}$$

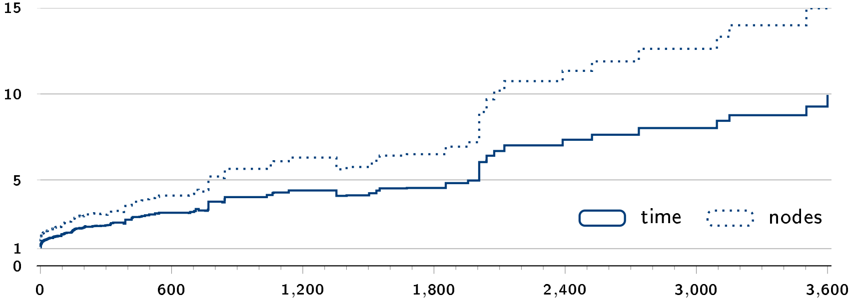


[See also Achterberg and Wunderling 2013]

Filtered Performance Diagrams

Gradually exclude instances **solved by A and B** and compute speedup:

$$t \mapsto \frac{\mu(\{N_{A,i} : \max\{t_{A,i}, t_{B,i}\} \geq t\})}{\mu(\{N_{B,i} : \max\{t_{A,i}, t_{B,i}\} \geq t\})}$$



[See also Achterberg and Wunderling 2013]

Outline

Analyzing MINLP solver components

Benchmarking methodology

Separation

Reformulation

Primal Heuristics

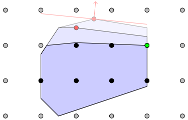
Tree search

Propagation

Separation

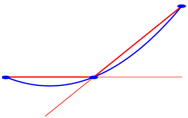
MIP cutting planes

- general: Gomory, cMIR, $\{0, \frac{1}{2}\}$-cuts, ...
- problem-specific: knapsack, clique, multi commodity flow, ...



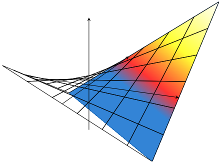
Gradient cuts for convex terms

- feasibility enforced without branching
- exploit integer information for univariate convex terms



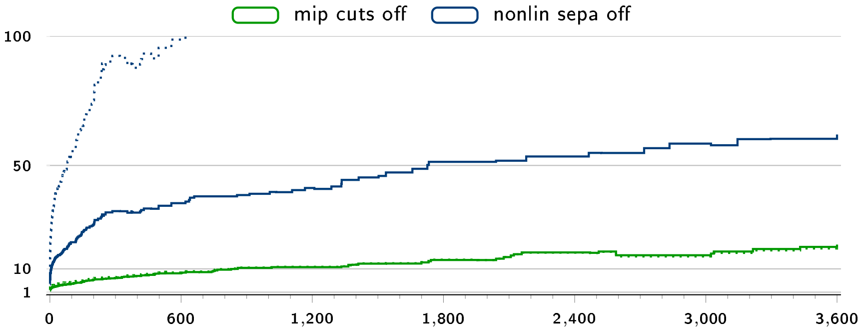
Convex underestimators for nonconvex terms

- secant, signed power, McCormick, ...



Alternative setting: off during fractional branching

Separation



		all		maxtime ≥ 100	
setting	solved	time	nodes	time	nodes
mip cuts off	−39	+65%	+107%	+333%	+395%
nonlin sepa off	−102	+302%	+695%	+1964%	+5569%

Outline

Analyzing MINLP solver components

Benchmarking methodology

Separation

Reformulation

Primal Heuristics

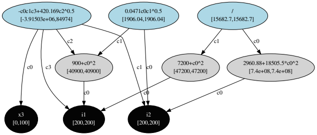
Tree search

Propagation

Reformulation

Expression graph reformulation

- ▷ decompose into Smith normal form
- ▷ identify common terms
- ▷ merge expressions, e.g., polynomials



Products with binary variables

- ▷ linearize using big-M

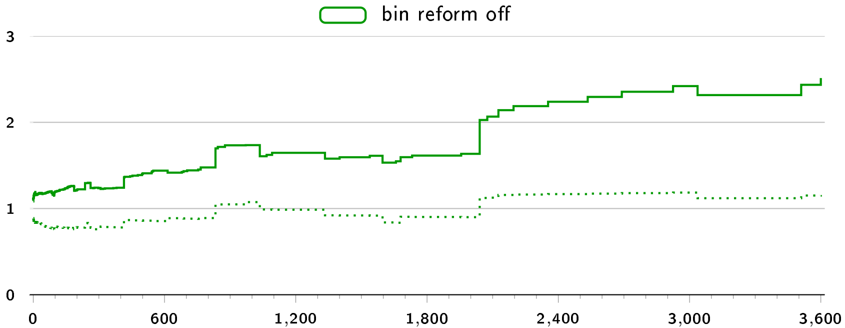
$$x \cdot \sum_k a_k y_k \quad \text{with} \quad x \in \{0,1\}$$

$$\downarrow$$

$$M^L x \le w \le M^U x,$$

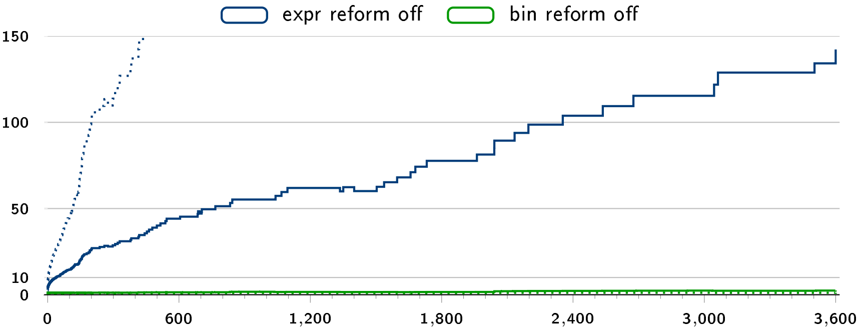
$$\sum_k a_k y_k - M^U(1-x) \le w \le \sum_k a_k y_k - M^L(1-x)$$

Reformulation



		all		maxtime $\geq$ 100	
setting	solved	time	nodes	time	nodes
expr reform off	−69	+160%	+322%	+1386%	+3631%
bin reform off	−9	+8%	−11%	+20%	−21%

Reformulation



		all		maxtime ≥ 100	
setting	solved	time	nodes	time	nodes
expr reform off	−69	+160%	+322%	+1386%	+3631%
bin reform off	−9	+8%	−11%	+20%	−21%

Outline

Analyzing MINLP solver components

Benchmarking methodology

Separation

Reformulation

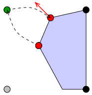
Primal Heuristics

Tree search

Propagation

Primal Heuristics

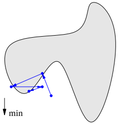
Besides waiting for feasible LP solutions . . .



Standard MIP heuristics applied to MIP relaxation

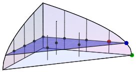
- ▷ rounding, diving, feasibility pump, . . .

NLP local search



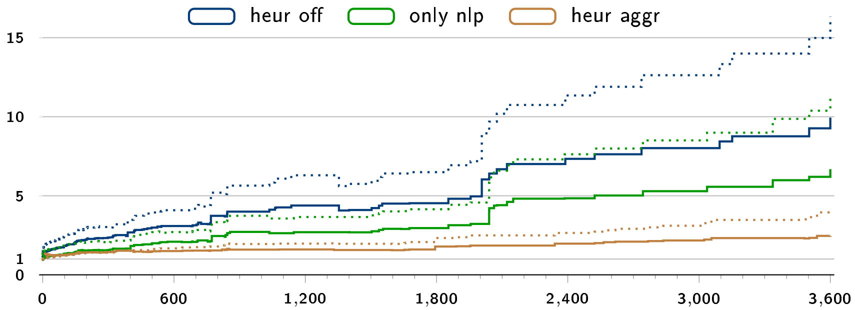
- ▷ for integer and LP feasible solutions
- ▷ fix integers and solve remaining NLP

MINLP heuristics



- ▷ NLP diving
- ▷ RENS [Berthold 2013]
- ▷ Undercover [Berthold and G. 2013]
- ▷ . . .

Primal Heuristics



		all		maxtime ≥ 100	
setting	solved	time	nodes	time	nodes
heur off	−19	+7%	+36%	+84%	+144%
only nlp	−11	−4%	+22%	+33%	+22%
heur aggr	−2	+27%	−4%	+28%	+86%

Analyzing MINLP solver components

Benchmarking methodology

Separation

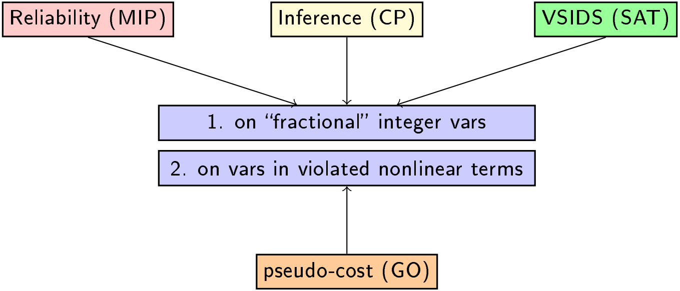
Reformulation

Primal Heuristics

Tree search

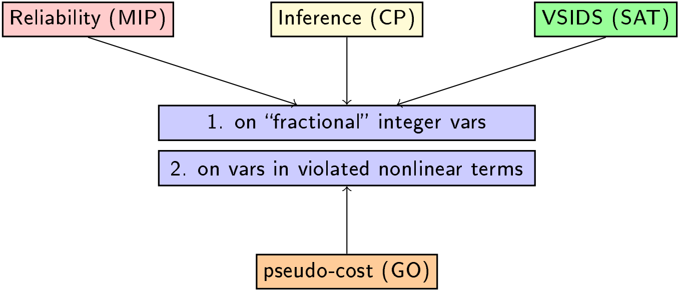
Propagation

Branching



Reliability (MIP)
Inference (CP)
VSIDS (SAT)
1. on "fractional" integer vars
2. on vars in violated nonlinear terms
pseudo-cost (GO)

Branching

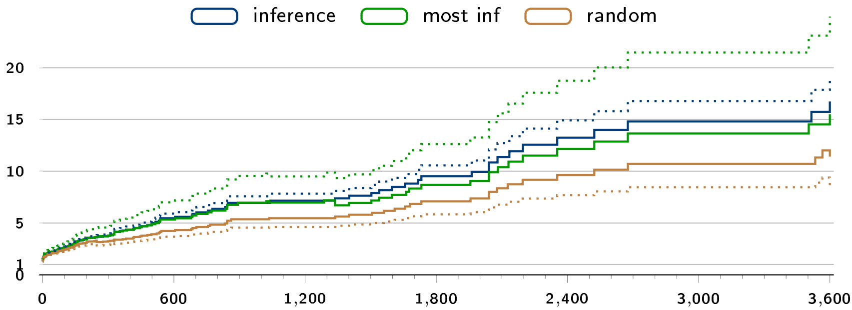


Alternative settings for spatial branching

- inference, most infeasible, random

[See Tawarmalani and Sahinidis 2002, Achterberg and Berthold 2009, Belotti et al. 2009, . . .]

Spatial Branching



		all		maxtime $\geq$ 100	
setting	solved	time	nodes	time	nodes
inference	−27	+31%	+34%	+167%	+176%
most inf	−24	+30%	+38%	+165%	+209%
random	−24	+30%	+28%	+145%	+130%

Node selection

Tasks

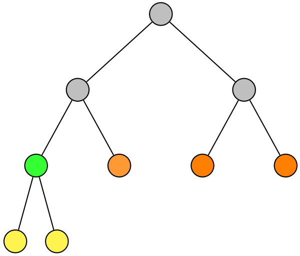
- improve primal bound
- keep computational effort small
- improve global dual bound

Best estimate with plunging

- select node Q with best/minimal (pseudo cost) estimate value for feasible solution quality

$$\bar{z}_Q + \sum_{k : \bar{x}_k \, fractional} \min\{\Psi^- f^-, \Psi^+ f^+\}$$

- plunge

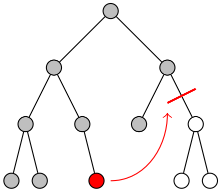


Alternative setting: breadth first search

Conflict analysis/nogood learning

Analyse reason for pruning a node

- branchings and propagations
- infeasible and bound exceeding LP relaxation: dual ray heuristic
- derive short nogoods/conflict constraints

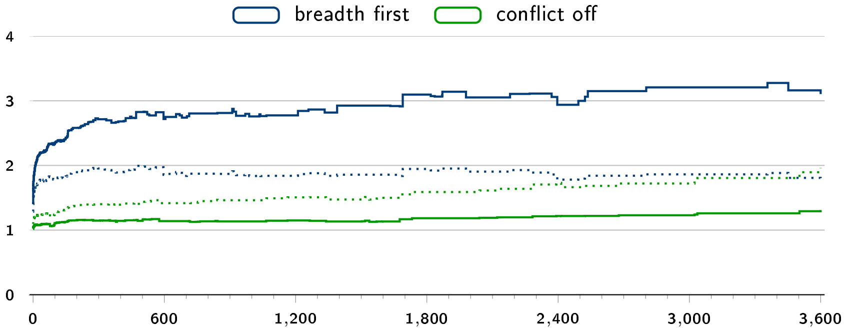


$$x_1 - x_3 \leq 0$$

Use subsequently

- to cut off other nodes
- to enable further propagations
- for VSIDS in branching

Node selection & conflict analysis



		all		maxtime ≥ 100	
setting	solved	time	nodes	time	nodes
breadth first	−22	+42%	+29%	+136%	+81%
conflict off	−2	+2%	+9%	+11%	+27%

Analyzing MINLP solver components

Benchmarking methodology

Separation

Reformulation

Primal Heuristics

Tree search

Propagation

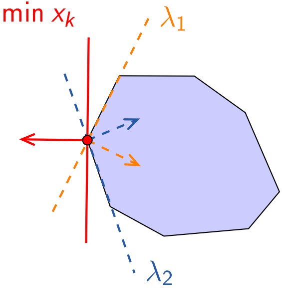
Bound tightening/propagation

Particularly important for nonconvex MINLP

- branching on continuous variables/infinite domains
- tight domains $\rightsquigarrow$ tight relaxation

Primal and dual reductions

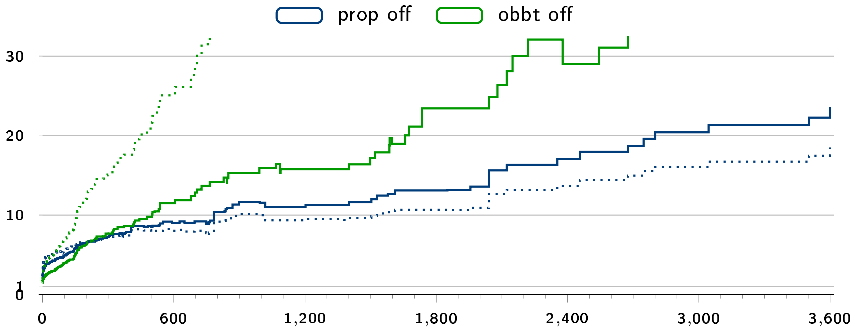
- reduced cost
- probing on binaries
- FBBT: feasibility-based bound tightening
- OBBT: optimization-based bound tightening and Lagrangian variable bounds:



$$x_k \geq \sum_{i:r_i>0} r_i \ell_i + \sum_{i:r_i<0} r_i u_i + \mu z^* + \lambda^T b$$

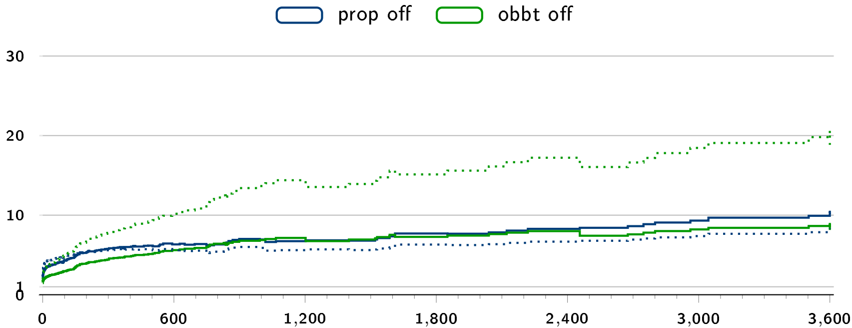
[Ryoo and Sahinidis 1996, Belotti et al. 2009, G. and Weltge 2013, ...]

Bound tightening/propagation



		all		maxtime ≥ 100	
setting	solved	time	nodes	time	nodes
prop off	−48	+90%	+129%	+397%	+461%
obbt off	−25	+47%	+93%	+303%	+607%

Bound tightening/propagation



		all		maxtime ≥ 100	
setting	solved	time	nodes	time	nodes
prop off	−48	+90%	+129%	+332%	+378%
obbt off	−25	+47%	+93%	+198%	+396%

Summary

		all		maxtime $\geq$ 100	
setting	solved	time	nodes	time	nodes
nonlin sepa off	−102	+302%	+695%	+1964%	+5569%
expr reform off	−69	+160%	+322%	+1386%	+3631%
prop off	−48	+90%	+129%	+397%	+461%
mip cuts off	−39	+65%	+107%	+333%	+395%
inference branching	−27	+31%	+34%	+167%	+176%
obbt off	−25	+47%	+93%	+303%	+607%
most inf branching	−24	+30%	+38%	+165%	+209%
random branching	−24	+30%	+28%	+145%	+130%
breadth first	−22	+42%	+29%	+136%	+81%
heur off	−19	+7%	+36%	+84%	+144%
heur only nlp	−11	−4%	+22%	+33%	+22%
bin reform off	−9	+8%	−11%	+20%	−21%
heur aggr	−2	+27%	−4%	+28%	+86%
conflict off	−2	+2%	+9%	+11%	+27%

Take-away messages

- SCIP is a global solver for nonconvex MINLPs
  open-source and free for academic research: `scip.zib.de`

Take-away messages

- SCIP is a global solver for nonconvex MINLPs
  open-source and free for academic research: `scip.zib.de`
- filtered performance diagrams

Take-away messages

- SCIP is a global solver for nonconvex MINLPs
  open-source and free for academic research: `scip.zib.de`
- filtered performance diagrams
- crucial components
  1. nonlinear separation
  2. expression graph reformulation
- add-on components
  1. propagation
  2. MIP cutting planes
  3. branching

Take-away messages

- SCIP is a global solver for nonconvex MINLPs
  open-source and free for academic research: `scip.zib.de`
- filtered performance diagrams
- crucial components
  1. nonlinear separation
  2. expression graph reformulation
- add-on components
  1. propagation
  2. MIP cutting planes
  3. branching

Thank you very much for your attention!