Measuring the Impact of Primal Heuristics

Timo Berthold

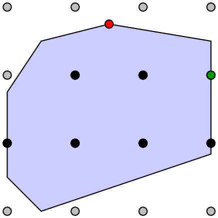
Zuse Institute Berlin

DFG Research Center MATHEON
Mathematics for key technologies

ISMP 2012, Berlin, 21/Aug/2012

Mixed-Integer Programming (MIP):

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & Ax \leq b \\ & x \in \mathbb{Z}^{n_I}_{\geq 0} \times \mathbb{R}^{n_C}_{\geq 0} \end{aligned}$$



Primal heuristics...

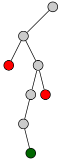
- ▷ are incomplete methods which
- ▷ often find good solutions
- ▷ within a reasonable time
- ▷ without any warranty!

Inside an exact solver...

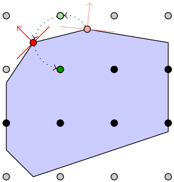
- ▷ they prove feasibility
- ▷ nearly optimal might be sufficient
- ▷ primal bound needed for pruning
- ▷ solutions guide remaining search

Categories of Heuristics

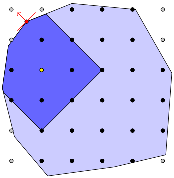
- Diving
  - simulate DFS with special branching rule
  - e.g., guided diving
  - one LP resolve (dual simplex) per iteration



- Objective diving
  - manipulate objective function
  - e.g., feasibility pump
  - one LP resolve (primal simplex) per iteration



- Large Neighborhood Search
  - solve sub-MIP
  - e.g., RINS, Local Branching
  - 500 nodes of a MIP



- Rounding, Propagation
  - no additional LPs or MIPs

How important are primal heuristics?

A major MIP software vendor says:

Our advanced MIP heuristics for quickly finding feasible solutions often produce good quality solutions where other solvers fall flat, leading to some of our biggest wins vs. the competition

How important are primal heuristics?

A major MIP software vendor says:

Our advanced MIP heuristics for quickly finding feasible solutions often produce good quality solutions where other solvers fall flat, leading to some of our biggest wins vs. the competition

Which heuristics?

- business secret
- few parameters to influence heuristics' behavior
- output only tells you that solution found by **some** heuristic

How important are primal heuristics?

A major MIP software vendor says:

Our advanced MIP heuristics for quickly finding feasible solutions often produce good quality solutions where other solvers fall flat, leading to some of our biggest wins vs. the competition

Which heuristics?

- business secret
- few parameters to influence heuristics' behavior
- output only tells you that solution found by **some** heuristic

$\rightsquigarrow$ **very important**

How important are primal heuristics?

Typical measure: Running time to prove optimality

How important are primal heuristics?

Typical measure: Running time to prove optimality

- one vendor: 6% improvement
- other vendor: 9% improvement
- non-commercial solver: 15 % improvement

How important are primal heuristics?

Typical measure: Running time to prove optimality

- ▷ one vendor: 6% improvement
- ▷ other vendor: 9% improvement
- ▷ non-commercial solver: 15 % improvement

$\rightsquigarrow$ **not important at all**

How important are primal heuristics?

Typical measure: Running time to prove optimality

- ▷ one vendor: 6% improvement
- ▷ other vendor: 9% improvement
- ▷ non-commercial solver: 15 % improvement

$\rightsquigarrow$ **not important at all**

So, what is wrong here?

How important are primal heuristics?

Typical measure: Running time to prove optimality

- ▷ one vendor: 6% improvement
- ▷ other vendor: 9% improvement
- ▷ non-commercial solver: 15 % improvement

$\rightsquigarrow$ **not important at all**

So, what is wrong here?

Goal of this talk: Introduce a new performance measure

How to measure the added value of a primal heuristic?

- time to optimality t_{solved}, number of branch-and-bound nodes
  - very much depends on dual bound

Comparing performance

How to measure the added value of a primal heuristic?

- time to optimality t_{solved}, number of branch-and-bound nodes
  - very much depends on dual bound
- time to best solution t_{opt}
  - nearly optimal solution might be found long before

How to measure the added value of a primal heuristic?

- time to optimality t_{solved}, number of branch-and-bound nodes
  - very much depends on dual bound
- time to best solution t_{opt}
  - nearly optimal solution might be found long before
- time to first solution t_1
  - disregards solution quality

Comparing performance

How to measure the added value of a primal heuristic?

- time to optimality t_{solved}, number of branch-and-bound nodes
  - very much depends on dual bound
- time to best solution t_{opt}
  - nearly optimal solution might be found long before
- time to first solution t_1
  - disregards solution quality
- performance profiles
  - depend on t_{solved}, hence on dual bound
  - not an absolute number

How to measure the added value of a primal heuristic?

- time to optimality t_{solved}, number of branch-and-bound nodes
  - very much depends on dual bound
- time to best solution t_{opt}
  - nearly optimal solution might be found long before
- time to first solution t_1
  - disregards solution quality
- performance profiles
  - depend on t_{solved}, hence on dual bound
  - not an absolute number
- primal integral

3 steps, 3 dates

3 steps we take on the next slides:

- primal gap
- primal gap function
- primal integral

3 pieces of information that we need:

- an optimal or best known solution $\tilde{x}_{opt}$
- development of incumbent solution (log file)
- the time limit t_{max}

Let $\tilde{x}$ be a solution, $\tilde{x}_{\text{opt}}$ be an optimum, $t_{\max} \in \mathbb{R}_{\geq 0}$ be a timelimit.

Primal gap $\gamma \in [0, 1]$ of $\tilde{x}$:

$$\gamma(\tilde{x}) := \begin{cases} 0, & \text{if } \left|c^T \tilde{x}_{\text{opt}}\right| = \left|c^T \tilde{x}\right| = 0, \\ 1, & \text{if } c^T \tilde{x}_{\text{opt}} \cdot c^T \tilde{x} < 0, \\ \frac{\left|c^T \tilde{x}_{\text{opt}} - c^T \tilde{x}\right|}{\max\{\left|c^T \tilde{x}_{\text{opt}}\right|, \left|c^T \tilde{x}\right|\}}, & \text{else.} \end{cases}$$

Primal gap function $p\colon [0, t_{\max}] \mapsto [0, 1]$:

$$p(t) := \begin{cases} 1, & \text{if no incumbent until point } t, \\ \gamma(\tilde{x}(t)), & \text{with } \tilde{x}(t) \text{ incumbent at point } t. \end{cases}$$

- step function, changes at points t_i when new incumbent found
- $p(0) = 1$, $p(t) = 0$ for all $t \geq t_{\text{opt}}$
- monotonously decreasing

Primal integral $P(T)$ of $T \in [0, \mathrm{t}_{\max}]$:

$$P(T) := \int_{t=0}^{T} p(t)\,dt = \sum_{i=1}^{I} p(t_{i-1}) \cdot (t_i - t_{i-1}),$$

Comparing performance

How to measure the added value of a primal heuristic?

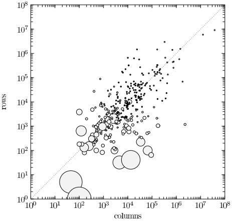
- time to optimality t_{solved}, number of branch-and-bound nodes
  - very much depends on dual bound
- time to best solution t_{opt}
  - nearly optimal solution might be found long before
- time to first solution t_1
  - disregards solution quality
- performance profiles
  - depend on t_{solved}, hence on dual bound
  - not an absolute number

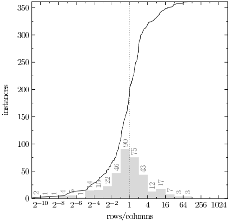
- primal integral $P(\mathrm{t}_{\max})$
  - favors finding good solutions early
  - considers each update of incumbent
  - $P(\mathrm{t}_{\max})/\mathrm{t}_{\max}$ "average solution quality"
  - expected quality of the incumbent, if stopped arbitrarily

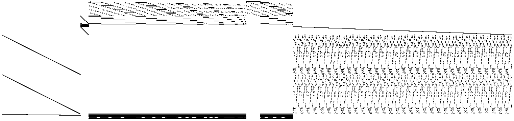
MIPLIB2010:

- 361 instances, benchmark set: 87
- 120–160k vars, 32–624k rows, 666–27M nz
- industry and academics
- diverse applications, combinatorics
- major vendors in comittee
- `http://miplib.zib.de`

- + MIPLIB2003, MIPLIB 3.0

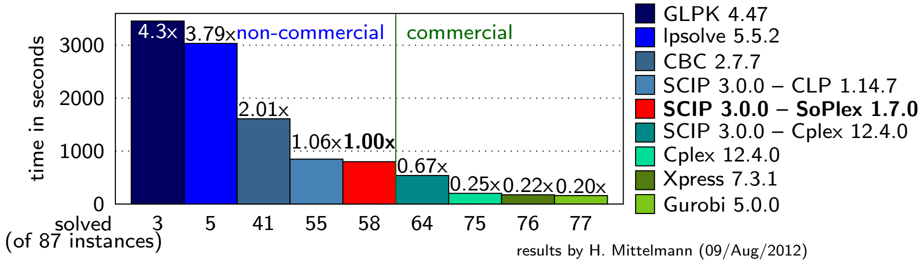






SCIP: Solving Constraint Integer Programs

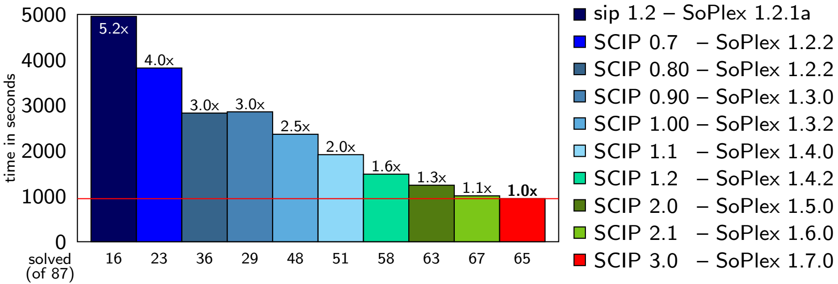
- standalone solver / branch-cut-and-price-framework
- modular structure via plugins
- free for academic use: `http://scip.zib.de`
- very fast non-commercial MIP solver



results by H. Mittelmann (09/Aug/2012)

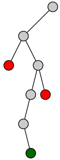
SCIP: Solving Constraint Integer Programs

- better support of MINLP
- new presolvers and propagators
- AMPL and MATLAB interface (beta)
- first releases of GCG and UG

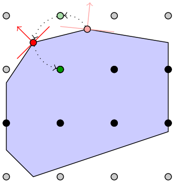



Primal Heuristics
linesearch diving
rootsol diving
intdiving
veclen diving
feaspump
pscost diving
objpscost diving
fracdiving
guided diving
cross over
dins
coef diving
under cover
rens
actcons diving
rins
subnlp
zi round
twoopt
local branching
trysol
trivial
fixand infer
mutation
shift& prop
octane
simple rounding
rounding
int shifting
shifting
oneopt

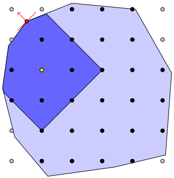
- Diving
  - simulate DFS with special branching rule
  - e.g., guided diving
  - one LP resolve (dual simplex) per iteration



- Objective diving
  - manipulate objective function
  - e.g., feasibility pump
  - one LP resolve (primal simplex) per iteration

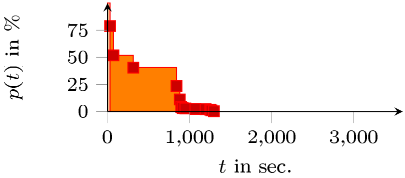


- Large Neighborhood Search
  - solve sub-MIP
  - e.g., RINS, Local Branching
  - 500 nodes of a MIP

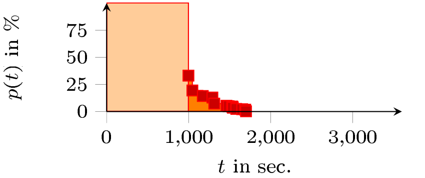


- Rounding, Propagation
  - no additional LPs or MIPs

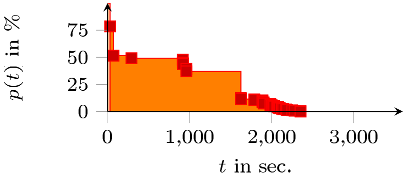
SCIP default, $P(\mathrm{t}_{\max}) = 421$



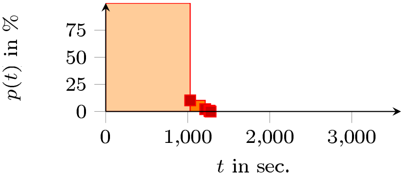
no round & prop, $P(\mathrm{t}_{\max}) = 1073$



only round & prop, $P(\mathrm{t}_{\max}) = 797$



no heuristics, $P(\mathrm{t}_{\max}) = 1050$



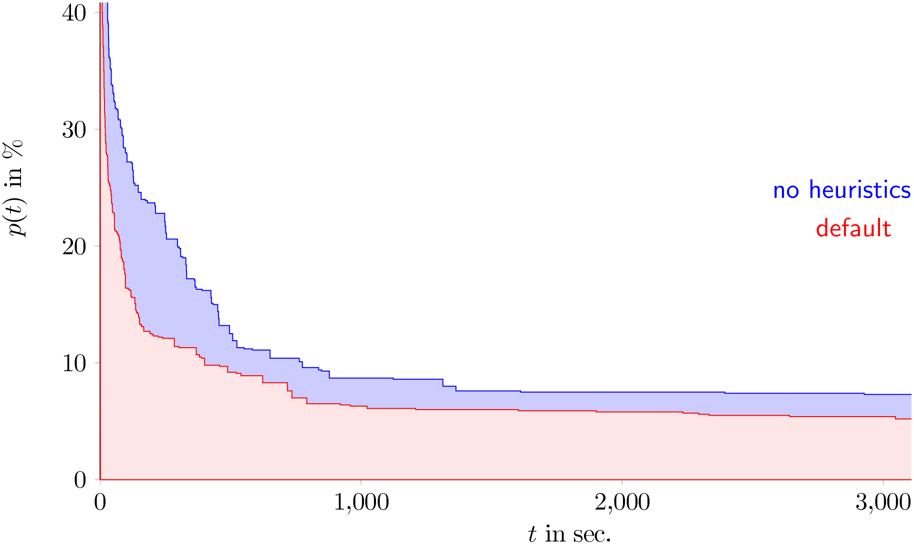
	def	noheur	nodive	noobj	nolns	noround
$\phi(t_1)$	3.8	15.4	4.1	3.9	4.0	6.1
$\phi(t_{opt})$	43.2	47.9	43.9	44.4	44.8	48.4
$\phi(t_{solved})$	107.5	114.7	109.7	114.8	110.2	105.9
$\phi(P(\mathrm{t_{max}}))$	257	363	299	277	275	263
$\phi(P(\mathrm{t_{max}}))/\mathrm{t_{max}}$	7.1%	10.1%	8.3%	7.7%	7.6%	7.3%

- primal heuristics extremely important for first solution
- rounding heuristics: slight degradation for time to optimality
- $P(\mathrm{t_{max}})$: def $\prec$ noround $\prec$ nolns $\approx$ nobj $\prec$ nodive $\prec$ noheur
- primal heuristics decrease average gap by more than 40%

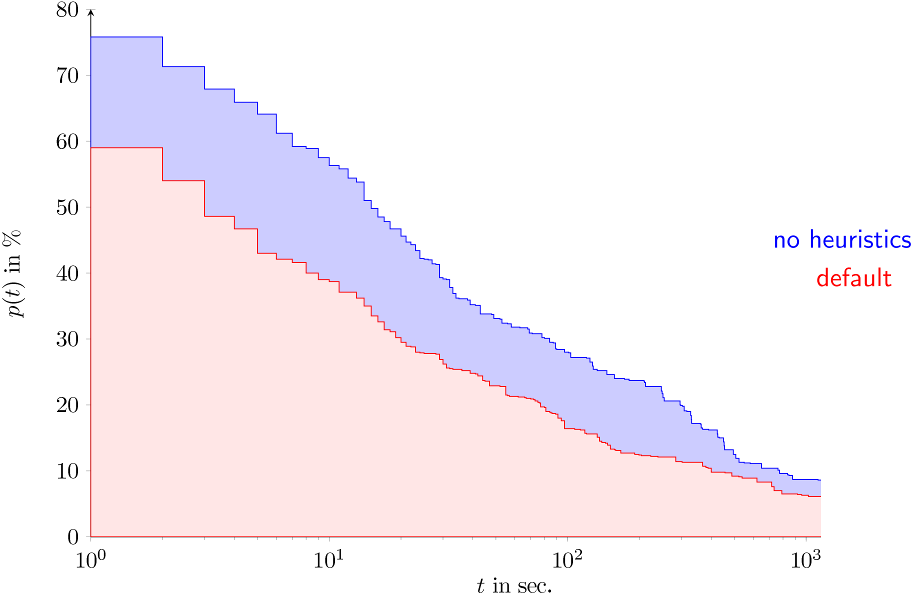
	def	noheur	dive	obj	lns	round
$\phi(t_1)$	3.8	15.4	9.7	7.2	11.8	4.8
$\phi(t_{\text{opt}})$	43.2	47.9	54.3	53.2	44.6	43.6
$\phi(t_{\text{solved}})$	107.5	114.7	115.3	108.6	110.5	112.9
$\phi(P(\mathrm{t}_{\max}))$	257	363	329	309	355	349
$\phi(P(\mathrm{t}_{\max}))/\mathrm{t}_{\max}$	7.1%	10.1%	9.1%	8.6%	9.9%	9.7%

- again, hardly any change in t_{opt} and t_{solved}
- rounding heuristics important for t_1
- $P(\mathrm{t}_{\max})$: single class cannot compensate the other

Average primal integral



Average primal integral (logarithmic)



Variants

variants of the primal integral:

- logarithmic time-axis (twice as early = twice as good)
- logarithmic gap-axis (twice as close to opt. = twice as good)
- consider dual gap (e.g. for cuts) or primal-dual gap
- consider other performance measures that change monotonously

Variants

variants of the primal integral:

- logarithmic time-axis (twice as early = twice as good)
- logarithmic gap-axis (twice as close to opt. = twice as good)
- consider dual gap (e.g. for cuts) or primal-dual gap
- consider other performance measures that change monotonously

future tests:

- test single primal heuristics
  - change SCIP defaults
  - which heuristics on which problems
- compare different solvers

Variants

variants of the primal integral:

- logarithmic time-axis (twice as early = twice as good)
- logarithmic gap-axis (twice as close to opt. = twice as good)
- consider dual gap (e.g. for cuts) or primal-dual gap
- consider other performance measures that change monotonously

future tests:

- test single primal heuristics
  - change SCIP defaults
  - which heuristics on which problems
- compare different solvers ...someone?

Primal integral:

- new performance measure
- captures overall solution process
- principle idea can be transfered to other measures

Measuring the impact:

- impact on time to optimality negligible
- overall impact (w.r.t. $P(\mathrm{t}_{\mathrm{max}})$) significant
- impact of single classes of heuristics limited

SCIP-related talks at ISMP

- *An exact rational mixed-integer programming solver*
  Kati Wolter, Wed.2.H0110
- *A generic branch-price-and-cut solver*
  Marco Lübbecke, Wed.3.H2032

- *Advances in linear programming*
  Matthias Miltenberger, Thu.3.H2033
- *LNS and diving heuristics in column generation algorithms*
  Christian Puchert, Thu.3.H2032
- *Approaches to solve mixed integer semidefinite programs*
  Sonja Mars, Thu.3.H2033

- *ParaSCIP and FiberSCIP – Parallel extensions of SCIP*
  Yuji Shinano, Fri.3.H1058
- *A computational comparison of symmetry handling methods in IP*
  Marc Pfetsch, Fri.3.H2013

Measuring the Impact of Primal Heuristics

Timo Berthold

Zuse Institute Berlin

DFG Research Center Matheon
Mathematics for key technologies