Using HiGHS as an LP solver within SCIP

Julian Hall

School of Mathematics
University of Edinburgh

jajhall@ed.ac.uk

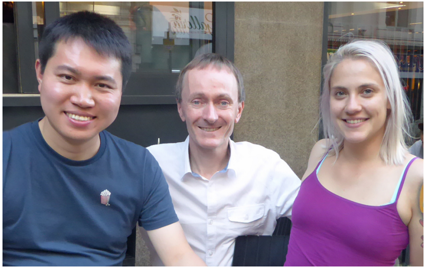
INFORMS Annual Meeting

Seattle

22 October 2019

HiGHS: A high-performance linear optimizer

HiGHS: **H**all, **i**vet **G**alabova, **H**uangfu, **S**chork (Feldmeier and Fogg)



Past (2011–2014)

- Written in C++ to study parallel simplex
- Dual simplex with standard algorithmic enhancements
- Efficient numerical linear algebra

Present (2016–date)

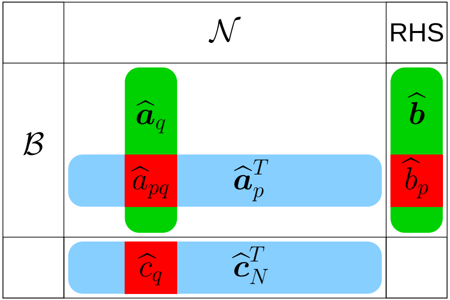
- Model management
- Interfaces
- Presolve
- Crash
- Interior point method

HiGHS: Dual simplex algorithm

Assume $\widehat{\mathbf{c}}_N \geq \mathbf{0}$ Seek $\widehat{\mathbf{b}} \geq \mathbf{0}$

Scan $\widehat{b}_i < 0$ for p to leave $\mathcal{B}$

Scan $\widehat{c}_j/\widehat{a}_{pj} < 0$ for q to leave $\mathcal{N}$



Update: Exchange p and q between $\mathcal{B}$ and $\mathcal{N}$

Update $\widehat{\mathbf{b}} := \widehat{\mathbf{b}} - \alpha_P \widehat{\mathbf{a}}_q$ $\alpha_P = \widehat{b}_p/\widehat{a}_{pq}$

Update $\widehat{\mathbf{c}}_N^T := \widehat{\mathbf{c}}_N^T + \alpha_D \widehat{\mathbf{a}}_p^T$ $\alpha_D = -\widehat{c}_q/\widehat{a}_{pq}$

Computation

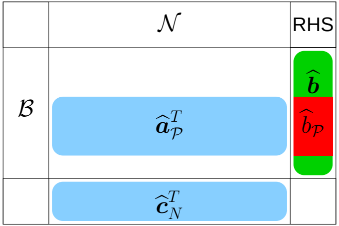
Pivotal row via $B^T \boldsymbol{\pi}_p = \mathbf{e}_p$ BTRAN and $\widehat{\mathbf{a}}_p^T = \boldsymbol{\pi}_p^T N$ PRICE

Pivotal column via $B\widehat{\mathbf{a}}_q = \mathbf{a}_q$ FTRAN Represent B^{-1} INVERT

Update B^{-1} exploiting $\bar{B} = B + (\mathbf{a}_q - B\mathbf{e}_p)\mathbf{e}_p^T$ UPDATE

HiGHS: Multiple iteration parallelism

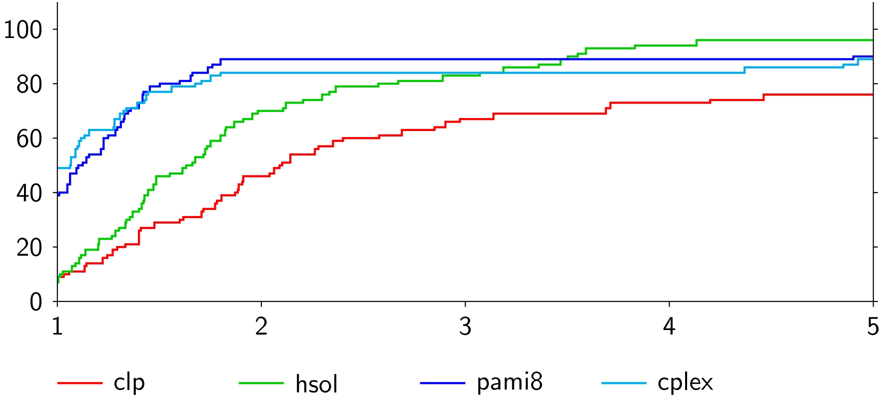
- Perform standard dual simplex minor iterations for rows in set $\mathcal{P}$ ($|\mathcal{P}| \ll m$)
- Suggested by Rosander (1975) but never implemented efficiently *in serial*



- Task-parallel multiple BTRAN to form $\boldsymbol{\pi}_{\mathcal{P}} = B^{-T}\mathbf{e}_{\mathcal{P}}$
- Data-parallel PRICE to form $\widehat{\mathbf{a}}_p^T$ (as required)
- Task-parallel multiple FTRAN for primal, dual and weight updates

Huangfu and H (2011–2014)
COAP best paper prize (2015)
MPC best paper prize (2018)

HiGHS: Performance



HiGHS: Benchmarks (9 October 2019)

Commercial

- Xpress
- Gurobi
- Cplex
- Mosek
- COPT
- Matlab
- QSopt
- SAS

Open-source

- Clp (COIN-OR)
- Glop (Google)
- Soplex (ZIB)
- Glpk (GNU)
- Lpsolve

Solver	COPT	Clp	SAS	Mosek	HiGHS	Glop	Matlab	Soplex
Time	1	1.4	3.2	3.7	5.4	7.2	8.1	10
Solved	40	40	37	38	37	35	32	36

Solver	QSopt	Glpk	Lpsolve
Time	26	28	108
Solved	34	31	23

SCIP: Solving Constraint Integer Programs

Requires: solver for LP sub-problems

$$\begin{array}{ll} \text{minimize} & f = \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u} \end{array}$$

Existing LP interfaces

- Commercial
  - `Cplex`
  - `Xpress`
  - `Gurobi`
  - `Mosek`
- Open source
  - `Soplex`
  - `Clp`
  - `QSopt`

Writing an interface

- What's needed?
- What's hard?

SCIP: Linear programming interface

What's needed?

Efficient solution of

$$\text{minimize} \quad f = \mathbf{c}^T\mathbf{x} \qquad \text{subject to} \quad A\mathbf{x} = \mathbf{b} \qquad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

when

- $\mathbf{c}$, $\mathbf{l}$ or $\mathbf{u}$ change
- Rows or columns are added to A
- Rows or columns are removed from A

Other standard utilities

For a solution with basis matrix B

- Form row i or column j of B^{-1}
- Form row i or column j of $B^{-1}A$

Advanced utilities

- Existence of rays of primal/dual unboundedness
- Strong branching

SCIP + HiGHS restart - simple view

"Branch on fractional variables"

After solving

$$\text{minimize} \quad f = \mathbf{c}^T\mathbf{x} \qquad \text{subject to} \quad A\mathbf{x} = \mathbf{b} \qquad \mathbf{l} \le \mathbf{x} \le \mathbf{u}$$

solve child LP with $l_j = u_j$

- Lose primal feasibility; retain dual feasibility $\Rightarrow$ use dual simplex method
- Dual simplex can continue iterating without having to regain feasibility

"Take child node from stack"

- Start simplex solver from saved optimal basis of parent
- Have to reinvert B and recompute $\widehat{\mathbf{b}}$ and $\widehat{\mathbf{c}}_N$

SCIP + HiGHS restart after modifying $\mathbf{c}$, $\mathbf{l}$ or $\mathbf{u}$

After solving

$$\text{minimize} \quad f = \mathbf{c}^T\mathbf{x} \qquad \text{subject to} \quad A\mathbf{x} = \mathbf{b} \qquad \mathbf{l} \le \mathbf{x} \le \mathbf{u}$$

When modifying $\mathbf{c}$

- Retain primal feasibility; lose dual feasibility $\Rightarrow$ use primal simplex?
- Primal simplex can iterate using representation of B^{-1} from dual simplex
- Lose dual simplex "edge weights" so maybe continue with "Phase 1" dual simplex

When modifying $\mathbf{l}$ and/or $\mathbf{u}$

- Lose primal feasibility; can lose dual feasibility
- Continue with (Phase 1) dual simplex

SCIP + HiGHS restart after adding or removing rows or columns

After adding a row

- Slack of new row is basic (primal infeasible)
- Basis matrix $B' = \begin{bmatrix} B & \mathbf{0} \\ \mathbf{b}^T & 1 \end{bmatrix}$ has $B'^{-1} = \begin{bmatrix} B^{-1} & \mathbf{0} \\ -\mathbf{b}^T B^{-1} & 1 \end{bmatrix}$
- Can operate with B'^{-1} by operating with B^{-1} and exploiting structure of B'
- Dual steepest edge weight w_i' for row i is

$$w_i' = \|\mathbf{e}_i^T B'^{-1}\|_2 = \begin{cases} w_i & i < m' \\ \left\| \begin{bmatrix} \mathbf{b}^T B^{-1} & 1 \end{bmatrix} \right\|_2 & i = m' \end{cases}$$

- Slacks of new columns are nonbasic (primal feasible - dual infeasible), so basis matrix unchanged
- When removing rows or columns, can maintain a basis and update an invertible representation of B, but harder

SCIP + HiGHS dual steepest edge weights

- Too expensive to save parent's weights
- Forming $w_i = \|B^{-1}\mathbf{e}_i\|_2$ requires solution of $B^T\widehat{\boldsymbol{\pi}_i} = \mathbf{e}_i$
  - Not expensive if B is triangular and many iterations are performed (Crash)
  - Possible trick (Davis)
    - Observe
    $$w_i^2 = \|\widehat{\boldsymbol{\pi}_i}\|_2^2 = (B^{-1}\mathbf{e}_i)^T(B^{-1}\mathbf{e}_i) = \mathbf{e}_i^T B^{-T} B^{-1}\mathbf{e}_i$$
    - Form $LL^T = B^T B$, then
    $$w_i^2 = \mathbf{e}_i^T(L^{-T}L^{-1}\mathbf{e}_i)$$
    - Solving for just one component of $L^{-T}L^{-1}\mathbf{e}_i$ is *very* fast with codes like `Cholmod`
  - May still be expensive!
  - Approximate weights obtained using incomplete Cholesky may be a fair compromise
- Could just use dual Devex!

SCIP + HiGHS interface development

- On 12 August: Initial set-up of HiGHS in SCIP
  - Immediate segfault when HiGHS compiled with debug
  - No fix found by Miltenberger + Galabova
- To 18 October: Work-around used SCIP (debug) + HiGHS (release)
  - No immediate segfault
  - Ran non-deterministically!
  - Still flagged up bugs allowing interface to be developed
  - Late effort fixed bug!
    - One week from Feldmeier
    - Few hours from Vigerske
- From 18 October: SCIP (debug) + HiGHS (debug) run deterministically!

SCIP + HiGHS results

	SCIP + Soplex			SCIP + HiGHS			HiGHS
Model	Objective	Nodes	Time	Objective	Nodes	Time	Speedup
flugpl	1201500	0	0.05	Correct	Same	0.02	2.50
p0548	8691	1	0.20	Correct	Same	0.16	1.25
rgn	82.2	1	0.18	Correct	Same	0.16	1.13
gt2	21166	1	0.03	Correct	98	0.17	0.18
blend2	7.6	868	2.88	7.7	1126	1.59	
egout	568	1	0.02	581	1	0.05	
enigma	0	809	0.86	1E+020	2665	1.19	
lseu	1120	149	0.91	1214	303	0.27	

Geomean speedup:

- 1.5 on the 3 problems solved correctly with the same node count
- 0.9 on the 4 problems solved correctly

- SCIP: High performance open source MIP solver
- HiGHS: High performance open source LP solver
- Should - will! - be a great match-up!

Slides: http://www.maths.ed.ac.uk/hall/INFORMS-SCIP19

SCIP: https://scip.zib.de/
HiGHS: http://www.highs.dev/

Q. Huangfu and J. A. J. Hall.
Novel update techniques for the revised simplex method.
Computational Optimization and Applications, 60(4):587–608, 2015.

Q. Huangfu and J. A. J. Hall.
Parallelizing the dual revised simplex method.
Mathematical Programming Computation, 10(1):119–142, 2018.