HiGHS: A High-performance Linear Optimizer

Julian Hall

School of Mathematics
University of Edinburgh

jajhall@ed.ac.uk

INFORMS Annual Meeting

Seattle

21 October 2019

THE UNIVERSITY *of* EDINBURGH

HiGHS

HiGHS: High performance linear optimization

- Linear optimization
  - Linear programming (LP)

$$\text{minimize} \quad \mathbf{c}^T\mathbf{x} \qquad \text{subject to} \quad A\mathbf{x} = \mathbf{b} \qquad \mathbf{x} \geq \mathbf{0}$$

  - Convex quadratic programming (QP)

$$\text{minimize} \quad \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \qquad \text{subject to} \quad A\mathbf{x} = \mathbf{b} \qquad \mathbf{x} \geq \mathbf{0}$$

    Q positive semi-definite
- High performance
  - Serial techniques exploiting sparsity in A
  - Parallel techniques exploiting multicore architectures

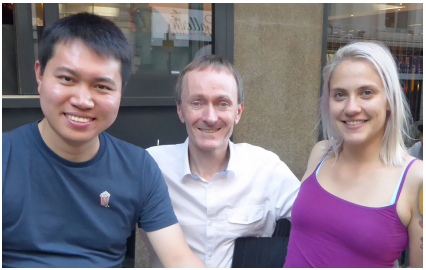
HiGHS: The team

What's in a name?

HiGHS: **H**all, **i**vet **G**alabova, **H**uangfu and **S**chork

Team HiGHS

- Julian Hall: Reader (1990–date)
- Ivet Galabova: PhD (2016–date)
- Qi Huangfu
  - PhD (2009–2013)
  - FICO Xpress (2013–2018)
- Lukas Schork: PhD (2015–2018)
- Michael Feldmeier: PhD (2018–date)
- Joshua Fogg: PhD (2019–date)



HiGHS: Dual simplex algorithm

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

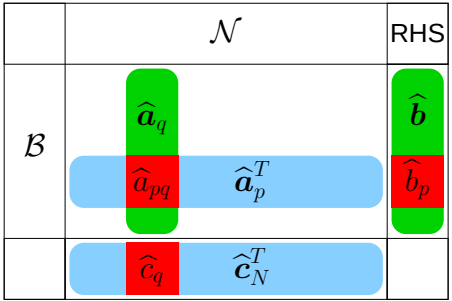
Scan $\widehat{b}_i < 0$ for p to leave $\mathcal{B}$

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

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



Computation

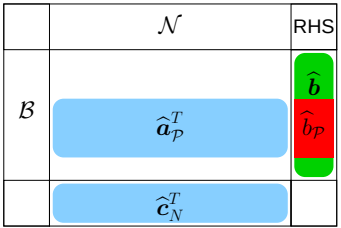
Pivotal row via $B^T\boldsymbol{\pi}_p = \mathbf{e}_p$ BTRAN and $\widehat{\mathbf{a}}_p^T = \boldsymbol{\pi}_p^T N$ PRICE

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: Present (2016–date)

Developments

- Model management: Load/add/delete/modify problem data
  Feldmeier, Galabova, H
- Interfaces
  Feldmeier, Galabova and Vigerske
- Presolve
  Galabova
- Crash
  Galabova and H
- Primal simplex solver
  Huangfu
- Interior point method
  Schork

HiGHS: Access

Source

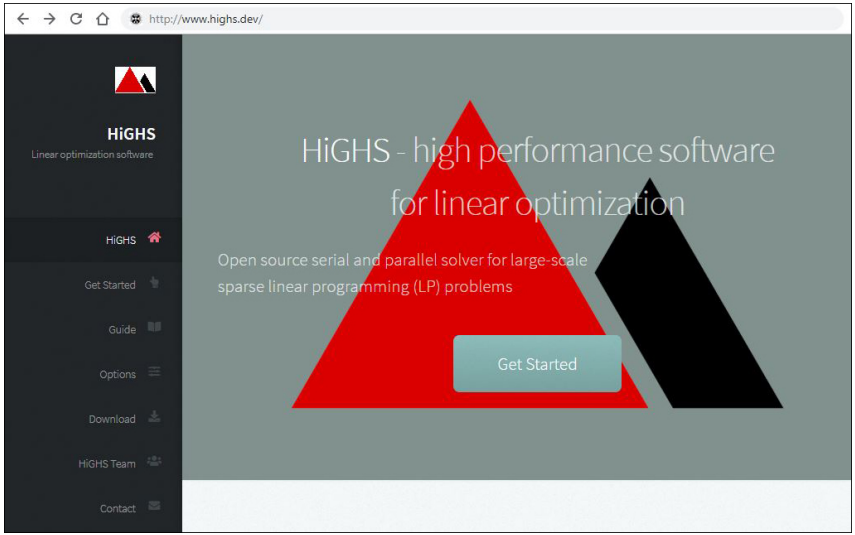
- Open source (MIT license)
- No third party code
- GitHub: ERGO-Code/HiGHS
- COIN-OR: Replacement for `Clp`?

Interfaces

- Existing
  - C++ `HiGHS` class
  - Load from `.mps`
  - Load from `.lp`
  - `C`
  - `C#`
  - `Julia`
  - `FORTRAN`
  - `Python`
  - `OSI` (almost!)
- Prototypes
  - `GAMS`
  - `SCIP`
- Planned
  - `AMPL`
  - `MATLAB`
  - `Mosel`
  - `PuLp`
  - `R`

  Suggestions?

HiGHS: http://www.highs.dev/



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

HiGHS: Comparison with Clp

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

Why is the HiGHS score so bad?

- HiGHS triangular crash not used
- HiGHS parallel code not used

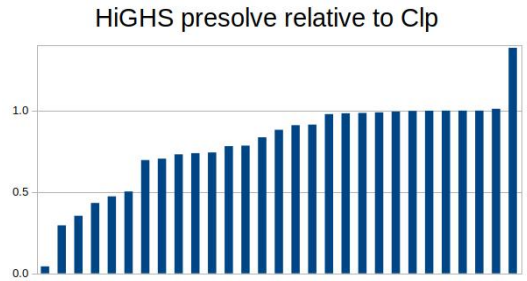
- Clp has a better presolve
- Clp has the Idiot crash
- Clp has a primal simplex solver

HiGHS: Presolve

Aim: eliminate rows, columns and nonzeros

Wide range of techniques

- Simple: interpret singleton rows as bounds on variables
- Complex: LP folding



Presolve measure

Product of

- Relative number of rows
- Relative number of columns
- Relative number of nonzeros

Presolve measure relative to Clp

- Better than Clp for 2/29 LPs!
- Within a factor 0.9 for 14/29 LPs
- Within a factor 0.7 for 23/29 LPs
- Within a factor 0.3 for 28/29 LPs
- Poor for one LP!

HiGHS: Tuning for the test problems

Everybody does it...

QAP problems

- Matrix entries $+1$ and -1, but $B = LU$ fills in
- Dual steepest edge weights become very inaccurate
- Monitor errors and switch to Devex
- Improves solution time for QAP15/NUG15 by a factor of 15!
- Contributes to reliability on general LPs?

HiGHS score now improved from 5.4 to ~4.4

HiGHS: Improvements

Solve all the problems!

- CONT11: Recognise when Markowitz tolerance ($u = 0.1$) needs to be increased
- IRISH-E: Add numerical pivot test to postsolve
- STP3D: Improve primal simplex "clean-up"

Immediate enhancements

- Add primal simplex solver (Huangfu)
- Resurrect `pami`
- Add triangular crash

HiGHS: The future

LP

- Improve presolve (Galabova)
- Add Idiot crash (Galabova)
- Add crossover (Feldmeier)
- Improved Idiot crash (Galabova)
- Direct linear system solver for IPM (Fogg)

MIP

- Publicise existing solver
- Improve existing solver

QP

- Active set QP solver (Feldmeier)
- IPM QP solver

Further interfaces

- `AMPL`
- `MATLAB`
- `Mosel`
- `PuLp`
- `R`

- High performance LP solver
- Reads: `.mps` and `.lp`
- Interfaces: C++ (native) `C`, `C#`, `Julia`, `FORTRAN`, `Python`
- Permissive licence and no third-party code
- Research and consultancy

Slides: http://www.maths.ed.ac.uk/hall/INFORMS-HiGHS19

HiGHS: http://www.highs.dev/

I. L. Galabova and J. A. J. Hall.
The "idiot" crash quadratic penalty algorithm for linear programming and its application to linearizations of quadratic assignment problems.
Optimization Methods and Software, April 2019. Published online.

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.

L. Schork and J. Gondzio.
Implementation of an interior point method with basis preconditioning.
Technical Report ERGO-18-014, School of Mathematics, University of Edinburgh, 2018.