HiGHS: a high-performance linear optimizer

Julian Hall

School of Mathematics, University of Edinburgh

2nd IMA and OR Society Conference on Mathematics of Operational Research

University of Aston, Birmingham

26 April 2019

THE UNIVERSITY *of* EDINBURGH

HiGHS

HiGHS: High performance linear optimization

- Linear optimization
  - Linear programming (LP)

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

  - Convex quadratic programming (QP)

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

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

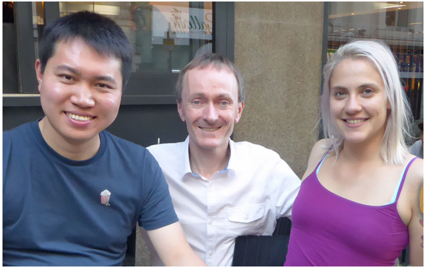
HiGHS: The team

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)
  - Google (2018)
- Qi Huangfu
  - PhD (2009–2013)
  - FICO Xpress (2013–2018)
  - MSc (2018–date)
- Lukas Schork: PhD (2015–2018)
- Michael Feldmeier: PhD (2018–date)



`HiGHS`: Past (2011–2014)

Overview

- Written in C++ to study parallel simplex
- Dual simplex with standard algorithmic enhancements
- Efficient numerical linear algebra
- No interface or utilities

Concept

- High performance serial solver (`hsol`)
- Exploit limited task and data parallelism in standard dual RSM iterations (`sip`)
- Exploit greater task and data parallelism via minor iterations of dual SSM (`pami`)

Huangfu and H

HiGHS: Dual simplex algorithm

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

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

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

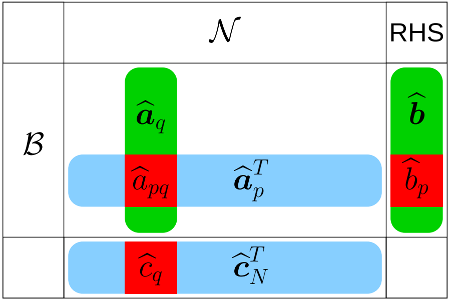
Update: Exchange p and q between $\mathcal{B}$ and $\mathcal{N}$

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

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

Data required

- Pivotal row $\widehat{\boldsymbol{a}}_p^T = \boldsymbol{e}_p^T B^{-1} N$
- Pivotal column $\widehat{\boldsymbol{a}}_q = B^{-1} \boldsymbol{a}_q$

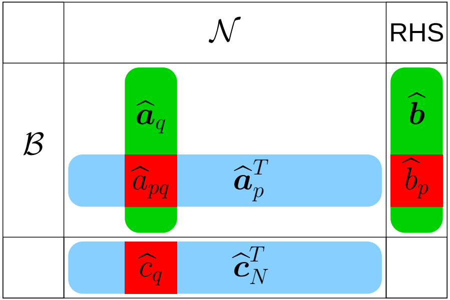


HiGHS: Dual simplex algorithm

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

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

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

Computation

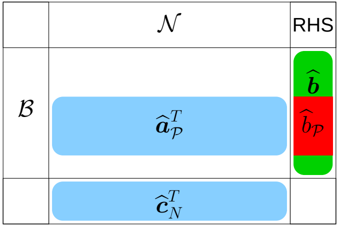
Pivotal row via $B^T \boldsymbol{\pi}_p = \boldsymbol{e}_p$ BTRAN and $\widehat{\boldsymbol{a}}_p^T = \boldsymbol{\pi}_p^T N$ PRICE

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

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

`HiGHS`: Multiple iteration parallelism with `pami` option

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

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

HiGHS: Performance and reliability

Extended testing using 159 test problems

- 98 Netlib
- 16 Kennington
- 4 Industrial
- 41 Mittelmann

Exclude 7 which are "hard"

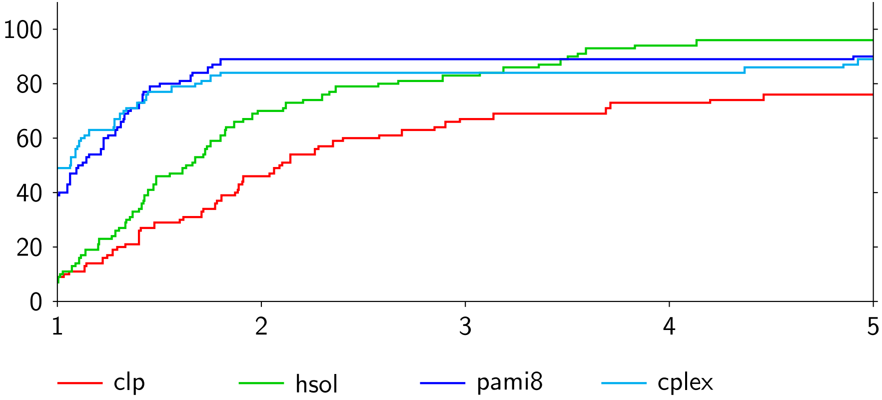
Performance

Benchmark against `clp` (v1.16) and `cplex` (v12.5)

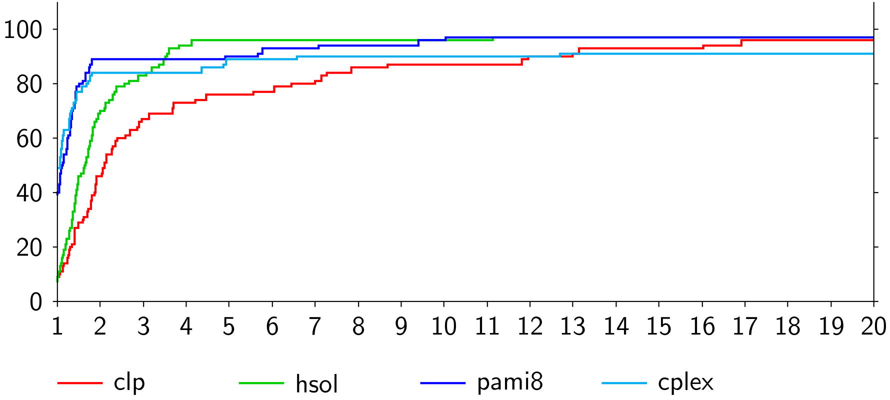
- Dual simplex
- No presolve
- No crash

Ignore results for 82 LPs with minimum solution time below 0.1s

HiGHS: Performance



HiGHS: Reliability



HiGHS: Present (2016–date)

Developments

- Model management: Load/add/delete/modify problem data

  Feldmeier, Galabova, H
- Interfaces

  Feldmeier, Galabova, Vigerske
- Presolve

  Galabova
- Crash

  H and Galabova
- Interior point method

  Schork

HiGHS: Access

Source

- Open source (MIT license)
- GitHub: ERGO-Code/HiGHS
- COIN-OR: Replacement for `Clp`?

Interfaces

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

- No more excuses!
- Use the 40 Mittelmann test LP problems
  - Some familiar - some not
  - Some easy - some not

	Rows	Cols	Nonzeros	$\frac{\text{Rows}}{\text{Cols}}$	$\frac{\text{Nonzeros}}{\text{Rows} \times \text{Cols}}$	$\frac{\text{Nonzeros}}{\max(\text{Rows},\text{Cols})}$
Min	960	1560	38304	1/255	0.0005%	2.2
Geomean	41074	60663	541193	0.6771	0.02%	5.1
Max	986069	1259121	11279748	85	16%	156.0

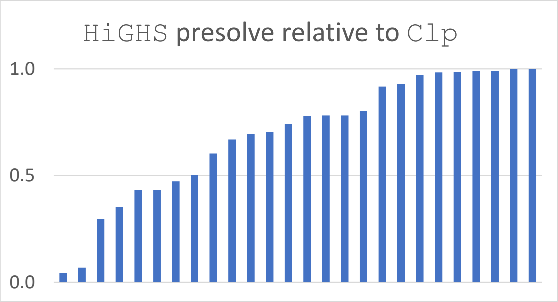
- Compete with other solvers in “vanilla” state

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

- Presolve valuable for 26/40 LPs
- Within a factor 0.9 for 9/26 LPs
- Within a factor 0.5 for 19/26 LPs

HiGHS: Crash

Aim: Identify basis more likely to be feasible

- Start with "all-slack" basis so $B = I$
- Perform basis changes
  - Aim to remove fixed slack
  - Replace with free/bounded/boxed structural
  - Maintain triangular B

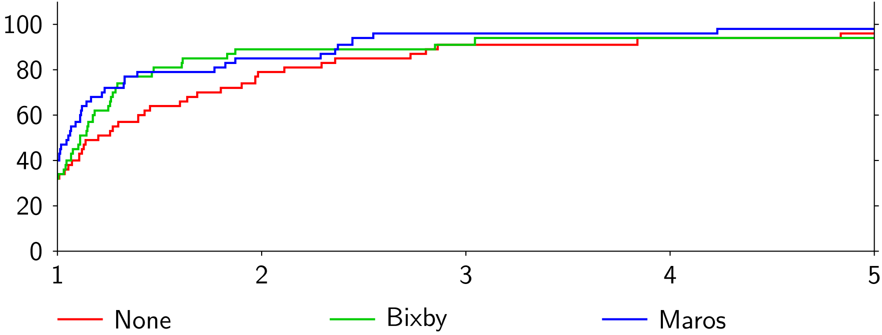
Bixby (1992)

- More aggressive crash also aims to
  - Replace boxed slack with free/bounded structural
  - Replace bounded slack with free structural

Maros and Mitra (1998)

- Value is problem-dependent
- Designed for primal simplex method
- Can it be valuable for dual simplex method?

HiGHS: Crash



- Bixby crash improved performance by 15%: best is a factor of 4.1
- Maros crash improved performance by 21%: best is a factor of 19.

HiGHS: Benchmarks

Commercial

- Xpress
- Gurobi
- Cplex
- Mosek
- SAS
- Matlab

Open-source

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

Solver	Clp	Mosek	SAS	HiGHS	Glop	Matlab	Soplex	Glpk	Lpsolve
Time	1	2.8	3.2	5.3	6.4	6.6	10.1	26	112

Mittelmann (17 Mar 2019)

Why is the HiGHS score so bad?

- HiGHS presolve not used
- HiGHS triangular crash not used
- HiGHS parallel code not used
- Clp has the Idiot crash
- Clp has a primal simplex solver

HiGHS: Benchmarks

Test set	Clp	HiGHS
Mittelmann (17 March 2019)	1	5.3
All 40 LPs (23 April 2019)	1	3.1
Less 7 LPs where Idiot crash aids Clp significantly	1	2.0
Less 14 LPs where Clp uses primal simplex	1	1.4
Remaining 19 LPs with HiGHS presolve	1.004	1

What's still to come with HiGHS?

- Resurrect pami
- Resurrect triangular crash
- Study 20 new test problems
- Improve presolve
- Add Idiot crash
- Add crossover
- Add primal simplex solver

HiGHS: The future is quadratic!

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

In particular

- $Q = A^T A$ and no equations

  Idiot crash for fast approximate solution of (some) LP problems (Galabova and H)
- $Q = I$

  Regularization terms in novel techniques for fast approximate solution of LP problems
- Q as Hessian of the Lagrangian?

  SQP methods for NLP

Solve efficiently (direct and/or iterative methods) for objective

$$f(\boldsymbol{x}) = \frac{\rho}{2}\boldsymbol{x}^T Q\boldsymbol{x} + \frac{\mu}{2}||A\boldsymbol{x} - \boldsymbol{b}||_2^2 + \frac{\nu}{2}||\boldsymbol{x} - \boldsymbol{x}_0||_2^2 + \boldsymbol{c}^T\boldsymbol{x}$$

Feldmeier (2018–date)

- High performance LP solver: simplex and interior point
- Interfaces: `OSI`, `SCIP`, `GAMS`
- Languages: `Python`, `FORTRAN`, `Julia`
- Research
- Consultancy

Slides:

http://www.maths.ed.ac.uk/hall/IMA-ORS19

Code:

https://github.com/ERGO-Code/HiGHS

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.