HiGHS
High-performance open-source software for linear optimization

Julian Hall
School of Mathematics
University of Edinburgh
jahall@ed.ac.uk

CO@Work 2020

21 September 2020
HiGHS: The team

What’s in a name?
HiGHS: Hall, Ivet Galabova, Huangfu and Schork

Team HiGHS
- Julian Hall: Reader (1990–date)
- Ivet Galabova: PhD (2016–date)
- Qi Huangfu
  - PhD (2009–2013)
- Michael Feldmeier: PhD (2018–date)
**HiGHS: Solvers**

**Linear programming (LP)**
- Dual simplex (Huangfu and Hall)
  - Serial techniques exploiting sparsity
  - Parallel techniques exploiting multicore architectures
- Interior point (Schork)
  - Highly accurate due to its iterative linear system solver
  - Crossover to a basic solution

**Mixed-integer programming (MIP)**
Prototype solver
HiGHS: Features and interfaces

Features
- Model management: Load/add/delete/modify problem data
- Presolve
- Crash

Interfaces

Language
- C++ HiGHS class
- Load from .mps
- Load from .lp
- C
- C#
- Julia
- FORTRAN
- Python

Applications
- GAMS
- JuliaOpt
- OSI
- SCIP
- SciPy

Future
- AMPL
- MATLAB
- Mosel
- PuLP
- R

Suggestions?
HiGHS: Access

- Open-source (MIT license)
  - GitHub: ERGO-Code/HiGHS
  - COIN-OR: Successor to Clp?
- No third-party code required
- Runs under Linux, Windows and Mac
- Build requires CMake 3.15
- Parallel code uses OpenMP
## HiGHS: Simplex benchmarks (14 August 2020)

<table>
<thead>
<tr>
<th>Solver</th>
<th>MindOpt</th>
<th>Gurobi</th>
<th>COPT</th>
<th>Clp</th>
<th>Mosck</th>
<th>SAS</th>
<th>HiGHS</th>
<th>Glop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>1</td>
<td>1.1</td>
<td>1.4</td>
<td>3.0</td>
<td>6.0</td>
<td>6.1</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>Solved</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solver</th>
<th>Matlab</th>
<th>Soplex</th>
<th>Glpk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>16</td>
<td>16</td>
<td>54</td>
</tr>
<tr>
<td>Solved</td>
<td>33</td>
<td>39</td>
<td>31</td>
</tr>
</tbody>
</table>

### Commercial
- COPT
- Gurobi
- Matlab
- MindOpt
- Mosk
- SAS

### Open-source
- Clp (COIN-OR)
- Glop (Google)
- Soplex (ZIB)
- Glpk (GNU)
HiGHS: Comparison with Clp

<table>
<thead>
<tr>
<th>Solver</th>
<th>MindOpt</th>
<th>Gurobi</th>
<th>COPT</th>
<th>Clp</th>
<th>Mosek</th>
<th>SAS</th>
<th>HiGHS</th>
<th>Glop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>1</td>
<td>1.1</td>
<td>1.4</td>
<td>3.0</td>
<td>6.0</td>
<td>6.1</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>Solved</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>35</td>
</tr>
</tbody>
</table>

Why is the HiGHS score so bad?

- HiGHS parallel code not used
- HiGHS triangular crash not used
- Clp has a better presolve
- Clp has the Idiot crash
- Clp has a primal simplex solver
- Clp has “sprint” (”sifting”) variant
HiGHS: SCIP Interface

Development

- Speed of HiGHS relative to Soplex motivates writing a HiGHS interface to SCIP
- Prototype interface now complete, but lacks
  - Primal ray for unbounded LPs
  - Farkas proof for infeasibility
  - Full efficiency when hot-starting nodes
- Passes 94% of the SCIP unit tests

Initial results: Time limit 3600s

<table>
<thead>
<tr>
<th></th>
<th>Solves</th>
<th>Timeout</th>
<th>Fails</th>
<th>Faster</th>
<th>M-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCIP-Soplex</td>
<td>89</td>
<td>5</td>
<td>1</td>
<td>41</td>
<td>105</td>
</tr>
<tr>
<td>SCIP-HiGHS</td>
<td>65</td>
<td>22</td>
<td>8</td>
<td>7</td>
<td>242</td>
</tr>
</tbody>
</table>

95 MIPLIB problems that Mittelmann’s SCIP-Soplex test solves in 1000s
HiGHS: The future

LP
- Improve presolve (Galabova)
- Add primal simplex solver and sifting (Hall)
- Improve simplex performance (Hall)
- Add Idiot crash and crossover (Galabova and Hall)
- Improve Idiot crash (Galabova)

QP
- Active set QP solver (Feldmeier)

MIP
- Develop successor to Cbc?

Further interfaces
- AMPL
- MATLAB
- Mosel
- PuLP
- R
High performance LP solvers: simplex and IPM
Reads: .mps and .lp
Language interfaces: C++, C, C#, Julia, FORTRAN, Python
Application interfaces: GAMS, JuliaOpt, OSI, SCIP, SciPy
Permissive license and no third-party code
Available for research and consultancy

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

I. L. Galabova and J. A. J. Hall.

Q. Huangfu and J. A. J. Hall.

Q. Huangfu and J. A. J. Hall.

L. Schork and J. Gondzio.