High performance computational techniques for the simplex method

Julian Hall

School of Mathematics
University of Edinburgh

jajhall@ed.ac.uk

CO@Work 2020

17 September 2024

THE UNIVERSITY of EDINBURGH

HiGHS

Overview

- Computational view of simplex algorithms
- Serial techniques
  - Hyper-sparsity
  - Cost perturbation
- Parallel techniques for general LP problems

Solving LP problems: Background

$$\text{minimize } \boldsymbol{c}^T\boldsymbol{x} \quad \text{such that} \quad A\boldsymbol{x} = \boldsymbol{b} \quad \text{and} \quad \boldsymbol{x} \geq \boldsymbol{0}$$

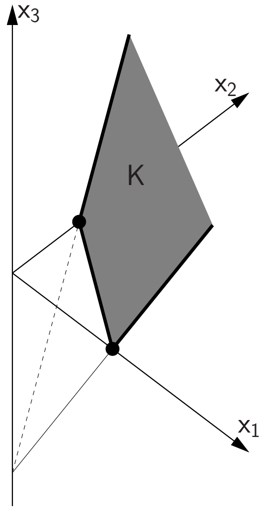
Background

- Fundamental model in optimal decision-making
- Solution techniques
  - Simplex method (1947)
  - Interior point methods (1984)
  - First order methods (2021)
- Large problems have
  - 10^3–10^8 variables
  - 10^3–10^8 constraints
- Matrix A is usually **sparse** and may be **structured**

Example

STAIR: 356 rows, 467 columns and 3856 nonzeros

Solving LP problems: Background



minimize $\boldsymbol{c}^T\boldsymbol{x}$ such that $A\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$

- A **vertex** of the **feasible region** $K \subset \mathbb{R}^n$ has
  - m **basic** components, $i \in \mathcal{B}$
  - $n - m$ zero **nonbasic** components, $j \in \mathcal{N}$
- A and $\boldsymbol{x}$ are partitioned according to $\mathcal{B} \cup \mathcal{N}$

$$B\boldsymbol{x}_B + N\boldsymbol{x}_N = \boldsymbol{b} \quad \Rightarrow \boldsymbol{x}_B = B^{-1}(\boldsymbol{b} - N\boldsymbol{x}_N) = \widehat{\boldsymbol{b}} - \widehat{N}\boldsymbol{x}_N$$

  since the **basis matrix** B is nonsingular
- Reduced objective is then $f = \widehat{f} + \widehat{\boldsymbol{c}}_N^T\boldsymbol{x}_N$, where $\widehat{f} = \boldsymbol{c}_B^T\widehat{\boldsymbol{b}}$ and $\widehat{\boldsymbol{c}}_N^T = \boldsymbol{c}_N^T - \boldsymbol{c}_B^T B^{-1}N$
- For $\boldsymbol{x}_N = \boldsymbol{0}$, partition yields an optimal solution if there is Primal feasibility $\widehat{\boldsymbol{b}} \geq \boldsymbol{0}$; Dual feasibility $\widehat{\boldsymbol{c}}_N \geq \boldsymbol{0}$

Simplex tableau

- **Reduced LP** corresponding to partition $\mathcal{B} \cup \mathcal{N}$ of $\{1, \ldots, n\}$ with B nonsingular is

$$\text{minimize } \widehat{f} + \widehat{\boldsymbol{c}}_N^T \boldsymbol{x}_N \quad \text{such that} \quad \boldsymbol{x}_B + \widehat{N}\boldsymbol{x}_N = \widehat{\boldsymbol{b}} \quad \text{and} \quad \boldsymbol{x} \geq \boldsymbol{0}$$

- Convenient to represent this in the **simplex tableau**

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\widehat{N}$	$\widehat{\boldsymbol{b}}$
	$\widehat{\boldsymbol{c}}_N^T$	

Primal simplex algorithm

Primal simplex algorithm: Choose a column

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

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

	$\mathcal{N}$	RHS
$\mathcal{B}$		
	$\widehat{c}_q$ $\widehat{\boldsymbol{c}}_N^T$	

Primal simplex algorithm: Choose a row

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

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

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

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\widehat{\boldsymbol{a}}_q$ $\widehat{a}_{pq}$	$\widehat{\boldsymbol{b}}$ $\widehat{b}_p$

Primal simplex algorithm: Update cost and RHS

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

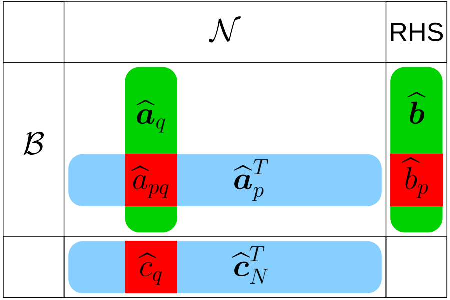
Scan $\widehat{c}_j < 0$ for q to leave $\mathcal{N}$

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

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

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

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



Primal simplex algorithm: Data required

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

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

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

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

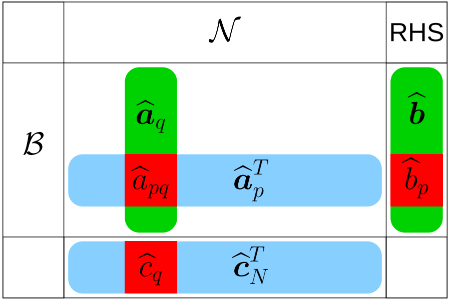
Update $\widehat{\boldsymbol{b}} := \widehat{\boldsymbol{b}} - \alpha_P \widehat{\boldsymbol{a}}_q$ $\alpha_P = \widehat{b}_p / \widehat{a}_{pq}$

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

Data required

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

Pivotal column $\widehat{\boldsymbol{a}}_q = B^{-1} \boldsymbol{a}_q$



Primal simplex algorithm: Revised simplex method computation

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

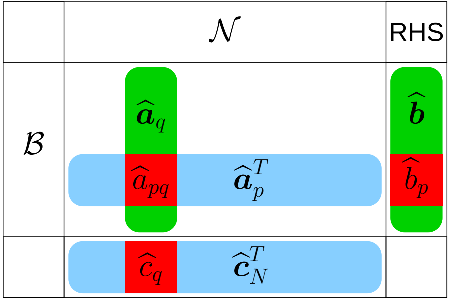
Scan $\widehat{c}_j < 0$ for q to leave $\mathcal{N}$

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

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 $\widehat{\boldsymbol{a}}_p^T = \boldsymbol{e}_p^T B^{-1} N$ via $B^T \boldsymbol{\pi}_p = \boldsymbol{e}_p$; $\qquad \widehat{\boldsymbol{a}}_p^T = \boldsymbol{\pi}_p^T N$

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

Dual simplex algorithm: Revised simplex method computation

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

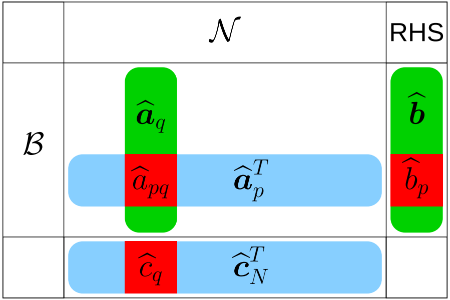
Scan $\widehat{b}_i < 0$ for p to leave $\mathcal{B}$

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 $\widehat{\boldsymbol{a}}_p^T = \boldsymbol{e}_p^T B^{-1} N$ via $B^T \boldsymbol{\pi}_p = \boldsymbol{e}_p$; $\qquad \widehat{\boldsymbol{a}}_p^T = \boldsymbol{\pi}_p^T N$

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

Solving LP problems: Primal or dual simplex?

Primal simplex algorithm

- Traditional variant
- Solution generally not primal feasible when (primal) LP is tightened

Dual simplex algorithm

- Preferred variant
- Easier to get dual feasibility
- More progress in many iterations
- Solution dual feasible when primal LP is tightened

In practice, both are required for efficiency and robustness

Simplex method: Computation

Standard simplex method (SSM): Major computational component

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\widehat{N}$	$\widehat{\boldsymbol{b}}$
	$\widehat{\boldsymbol{c}}_N^T$	

Update of tableau: $\widehat{N} := \widehat{N} - \dfrac{1}{\widehat{a}_{pq}}\widehat{\boldsymbol{a}}_q\widehat{\boldsymbol{a}}_p^T$

where $\widehat{N} = B^{-1}N$

- Hopelessly inefficient for sparse LP problems
- Prohibitively expensive for large LP problems

Revised simplex method (RSM): Major computational components

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

Mittelmann LP test set (2020)

Industry standard set of 40 LP problems

	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	54256	72442	910993	0.75	0.02%	6.5
Max	986069	1259121	11279748	85	16%	218.0

Mittelmann measure for solvers

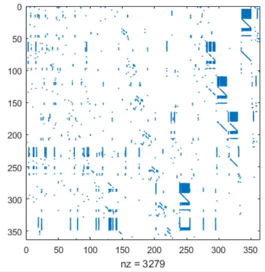
- Unsolved problems given "timeout" solution time
- Shift all solution times up by 10s
- Compute geometric mean of logs of shifted times
- **Solution time measure** is exponent of geometric mean shifted down by 10s
- **Mittelmann measure** for a solver is its solution time measure relative to the best

Hyper-sparsity: Solve $B\boldsymbol{x} = \boldsymbol{r}$ for sparse $\boldsymbol{r}$

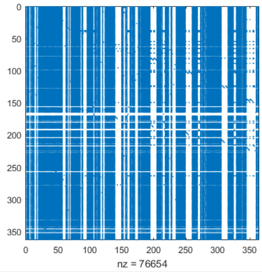
- In simplex, RHS of linear systems is sparse: column from A or unit vector
- When $\boldsymbol{r}$ is sparse, solution $\boldsymbol{x} = B^{-1}\boldsymbol{r}$ combines a few columns of B^{-1}
- Although B^{-1} is never formed explicitly, studying it is instructive

Inverse of a sparse matrix and solution of $B\boldsymbol{x} = \boldsymbol{r}$

Optimal B for LP problem STAIR



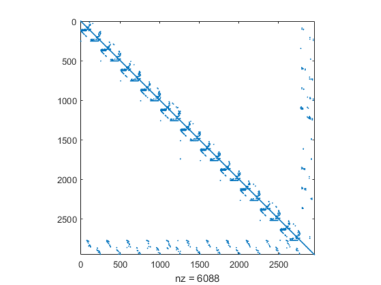
$B^{-1}\boldsymbol{r}$ is typically dense



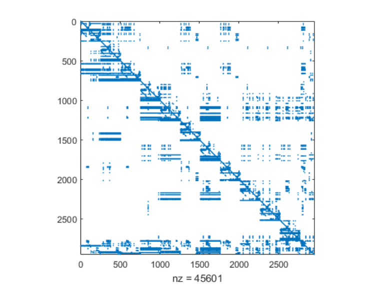
Hyper-sparsity: Solve $B\boldsymbol{x} = \boldsymbol{r}$ for sparse $\boldsymbol{r}$

Inverse of a sparse matrix and solution of $B\boldsymbol{x} = \boldsymbol{r}$

Optimal B for LP problem PDS-02



$B^{-1}\boldsymbol{r}$ is typically sparse



- If B^{-1} is sparse, then the LP is said to be **hyper-sparse**
- Huge performance gains from exploiting this in the simplex method

H and McKinnon (2005)

Hyper-sparsity: Effectiveness

Testing environment

- Mittelmann test set of 40 LPs
- `HiGHS` dual simplex solver with/without exploiting hyper-sparsity
- Time limit of 10,000 seconds

Results

- When exploiting hyper-sparsity: solves 37 problems
- When not exploiting hyper-sparsity: solves 34 problems

	Min	Geomean	Max
Iteration count increase	0.75	1.08	3.17
Solution time increase	0.83	2.31	67.13
Iteration speed decrease	0.92	2.14	66.43
Mittelmann measure		2.57	

Dual simplex: Cost perturbation

Dual degeneracy

- If some nonbasic dual values $\boldsymbol{c}_N^T - \boldsymbol{c}_B^T B^{-1} N$ are zero, the vertex is **dual degenerate**
- At a dual degenerate vertex, an iteration of the dual simplex algorithm may not lead to a strict increase in the dual objective
- Stalling or cycling may occur

Cost perturbation

- Add a small random value to some/all of the cost coefficients $\boldsymbol{c}$
- Nonbasic dual values then (at worst) take small positive values
- An iteration of the dual simplex algorithm yields (at least) a small positive increase in the dual objective
- When optimal, remove perturbations
- May require primal simplex iterations to regain optimality

Dual simplex: Cost perturbation - effectiveness

Results using Mittelmann test set

- With cost perturbation: `HiGHS` solves 37/40 problems
- Without cost perturbation: solves 27 problems

	Min	Geomean	Max
Iteration count increase	0.80	1.36	7.21
Solution time increase	0.57	1.46	13.31
Iteration speed decrease	0.49	1.07	4.02
Mittelmann measure		3.80	

Parallel simplex for general LP problems

Past work

- High value problem: many attempts, but almost nothing of practical value
  - Parallel tableau simplex: "easy" but useless
  - Parallel `PRICE` $\pi_p^T N$: "easy" but Amdahl reigns unless $m \ll n$
- Crazy asynchronous schemes: H and McKinnon (mid-90s)

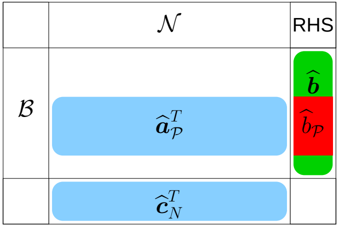
State-of-the-art

- High performance serial dual simplex solver with standard algorithmic enhancements (`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

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

Huangfu and H (2011–2014)

pami: Effectiveness

Serial overhead of pami

- HiGHS pami solver in serial: solves 34/40 problems

	Min	Geomean	Max
Iteration count increase	0.43	1.02	2.98
Solution time increase	0.31	1.62	5.36
Iteration speed decrease	0.69	1.59	5.11
Mittelmann measure		2.08	

Parallel speed-up of pami with 8 threads

	Min	Geomean	Max
Iteration count decrease	1.00	1.00	1.00
Solution time decrease	1.15	1.88	2.39
Iteration speed increase	1.15	1.88	2.39

pami: Effectiveness

Performance enhancement using parallel pami with 8 threads

	Min	Geomean	Max
Iteration count decrease	0.34	0.98	2.34
Solution time decrease	0.34	1.16	6.44
Iteration speed increase	0.38	1.18	2.75
Mittelmann measure		1.21	

Observations

- There is significant scope to improve pami performance further
- Use pami tactically: switch it off if it is ineffective

Commercial impact

- Huangfu applied the parallel dual simplex techniques within the Xpress solver
- For much of 2013–2018 the Xpress simplex solver was the best in the world

PDLP: A first order method for solving LPs

- PDLP is a new method for solving

$$\text{minimize} \quad f = \boldsymbol{c}^T\boldsymbol{x} \quad \text{such that} \quad A\boldsymbol{x} = \boldsymbol{b};\;\; \boldsymbol{x} \geq \boldsymbol{0}$$

  - Finds a saddle-point of $\min_{x \geq 0} \max_{y \geq 0} L(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{y}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y}$
  - Uses primal-dual hybrid gradient Chambolle-Pock (2011) update

$$\boldsymbol{x}^{t+1} = [\boldsymbol{x}^t - \tau(\boldsymbol{c} - A^T\boldsymbol{y}_t)]^+; \qquad \boldsymbol{y}^{t+1} = [\boldsymbol{y}^t + \mu A(2\boldsymbol{x}^{t+1} - \boldsymbol{x}^t)]^+$$

- Google's implementation in C++ on CPU is in the Mittelmann benchmarks (Applegate, Hinder, Lu, and Lubin – 2021)
- Better results for `cuPDLP-C` implementation in C+CUDA on GPU (Lu, Yang, Hu, Huangfu, Liu, Liu, Ye, Zhang – Dec 2023)
  - Available under MIT license on `GitHub` using `HiGHS` for file reading and presolve
    - CPU/GPU version `COPT` v7.0
    - CPU version in `HiGHS` v1.7.0

(March 2024)

HiGHS: Open-source software for large-scale sparse linear optimization

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

$$\text{minimize} \quad f = \frac{1}{2}\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \quad \text{such that} \quad A\boldsymbol{x} = \boldsymbol{b};\ \boldsymbol{x} \geq \boldsymbol{0}, \quad x_i \in \mathbb{Z},\ \forall i \in \mathcal{I}$$

Features

- Simplex, interior point and first order solvers for LP
- Branch-and-cut solver for MIP
- Active set solver for QP
- Written in C++
- Interfaces to other languages and systems

Availability

- Open-source (MIT license)
- No third-party code
- https://HiGHS.dev/

The world's best open-source linear optimization software

Mittelmann (2022–date)

Summary

Practical LP solution

Must exploit sparsity (and maybe structure)

- Simplex method
- Interior point methods
- First order methods

High performance simplex

- Best when solving families of related problems (MIP; SLP)
- Many (more) algorithmic and computational tricks in serial
- Parallel simplex has some impact on performance

D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy.
Practical large-scale linear programming using primal-dual hybrid gradient.
Advances in Neural Information Processing Systems, 34:20243–20257, 2021.

J. A. J. Hall and K. I. M. McKinnon.
Hyper-sparsity in the revised simplex method and how to exploit it.
Computational Optimization and Applications, 32(3):259–283, December 2005.

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