Novel update techniques for the revised simplex method

Qi Huangfu[1] Julian Hall[2]

[1]FICO

[2]School of Mathematics, University of Edinburgh

EUROPT

In memory of Roger Fletcher

13 July 2017

Overview

- Revised simplex method
  - Classical update techniques
- Novel update techniques
- Applications

LP and high performance simplex solvers

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

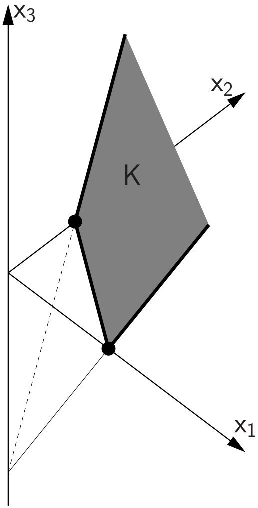
Background

- Fundamental model in optimal decision-making
- Simplex method preferred when solving related problems
- High performance requires
  - Algorithmic tricks
  - Computational tricks

Example

STAIR: 356 rows, 467 columns and 3856 nonzeros

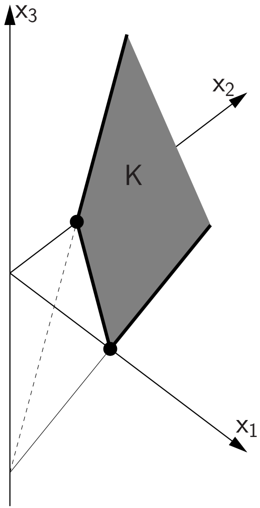
Solving LP problems: Characterizing the feasible region



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

- Equations partitioned according to $\mathcal{B} \cup \mathcal{N}$ as
$$B\boldsymbol{x}_B + N\boldsymbol{x}_N = \boldsymbol{b}$$
with nonsingular **basis matrix** B
- Solution set characterized by $\boldsymbol{x}_B = B^{-1}(\boldsymbol{b} - N\boldsymbol{x}_N)$
- At a **vertex** of K, $\begin{bmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{b}} \\ \boldsymbol{0} \end{bmatrix}$ where $\widehat{\boldsymbol{b}} = B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$
- K is $\boldsymbol{x}_B = \widehat{\boldsymbol{b}} - B^{-1}N\boldsymbol{x}_N$ for some $\boldsymbol{x}_N \geq \boldsymbol{0}$

Solving LP problems: Optimality conditions



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

- Objective partitioned according to $\mathcal{B} \cup \mathcal{N}$ as

$$f = \boldsymbol{c}_B^T \boldsymbol{x}_B + \boldsymbol{c}_N^T \boldsymbol{x}_N = \widehat{f} + \widehat{\boldsymbol{c}}_N^T \boldsymbol{x}_N$$

  where
  - $\widehat{f} = \boldsymbol{c}_B^T \widehat{\boldsymbol{b}}$
  - $\widehat{\boldsymbol{c}}_N^T = \boldsymbol{c}_N^T - \boldsymbol{c}_B^T B^{-1} N$ is the vector of **reduced costs**

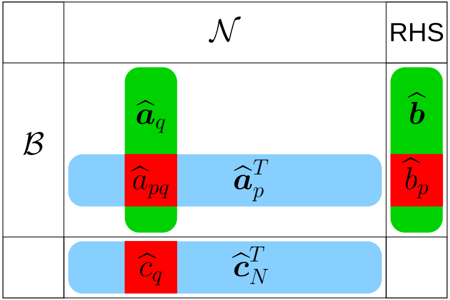
- Partition yields an optimal solution if there is
  - Primal feasibility $\widehat{\boldsymbol{b}} \geq \boldsymbol{0}$
  - Dual feasibility $\widehat{\boldsymbol{c}}_N \geq \boldsymbol{0}$

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



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

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

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 $\widehat{\boldsymbol{a}}_p^T = \boldsymbol{e}_p^T B^{-1} N$ via $B^T \boldsymbol{\pi}_p = \boldsymbol{e}_p$ $(\texttt{BTRAN})$; $\widehat{\boldsymbol{a}}_p^T = \boldsymbol{\pi}_p^T N$ $(\texttt{PRICE})$
- Pivotal column $\widehat{\boldsymbol{a}}_q = B^{-1} \boldsymbol{a}_q$ via $B \widehat{\boldsymbol{a}}_q = \boldsymbol{a}_q$ $(\texttt{FTRAN})$

Revised simplex method (RSM): Solving linear systems

- **Each iteration:**
  - Solve $B^T \boldsymbol{\pi}_p = \boldsymbol{e}_p$
  - Solve $B\,\widehat{\boldsymbol{a}}_q = \boldsymbol{a}_q$
  - Column p of B replaced by $\boldsymbol{a}_q$ to give $\bar{B} = B + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T$
- **Sparsity-exploiting decomposition:** $PBQ = LU$
- **Challenge:** Solve systems involving $\bar{B}$ at minimal cost

Simplex method: Product form update (PFI) for B

- Given

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

- Take B out as a factor on the left

$$\bar{B} \quad = \quad B[I + (B^{-1}\boldsymbol{a}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T] \quad = \quad BE$$

$$\text{where} \quad E \quad = \quad I + (\widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T \quad = \quad \begin{bmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & \\ & & \mu & & \\ & & \vdots & \ddots & \\ & & \eta_m & & 1 \end{bmatrix}$$

$\mu = \widehat{a}_{pq}$ is the **pivot**; remaining entries in $\widehat{\boldsymbol{a}}_q$ form the **eta vector** $\boldsymbol{\eta}$

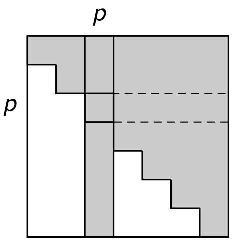
- Can solve $\bar{B}\boldsymbol{x} = \boldsymbol{r}$ as $B\boldsymbol{x} = \boldsymbol{r}$ then $\boldsymbol{x} := E^{-1}\boldsymbol{x}$ as

$$x_p := x_p/\mu \qquad \text{then} \qquad \boldsymbol{x} := \boldsymbol{x} - x_p\boldsymbol{\eta}$$

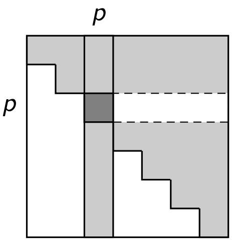
Dantzig and Orchard-Hays (1954)

Simplex method: Forrest-Tomlin update (FT) for B

- Given
$$\bar{B} = B + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T \quad \text{where (wlog)} \quad B = LU$$
- Multiply $\bar{B}$ by L^{-1} to give
$$L^{-1}\bar{B} \;=\; U + (L^{-1}\boldsymbol{a}_q - U\boldsymbol{e}_p)\boldsymbol{e}_p^T \;=\; U + (\widetilde{\boldsymbol{a}}_q - \boldsymbol{u}_p)\boldsymbol{e}_p^T \;=\; U' \qquad \text{(a)}$$
- Eliminate entries in row p to give $R^{-1}U' = \bar{U}$ (b)



(a) Spiked upper: U'



(b) After elimination: $\bar{U}$

- Yields $\bar{B} = LR\bar{U}$
- Compute $\widetilde{\boldsymbol{a}}_q$ when forming $\widehat{\boldsymbol{a}}_q$
- Represent R like E
- FT more efficient than PFI with respect to sparsity

Forrest and Tomlin (1972)

Simplex method: Multiple updates

- Suppose $B_0 = L_0 U_0$ and k updates are performed to obtain B_k
  - Pivot in rows $\{p_i\}_{i=1}^k$
  - Introduce columns $\{\boldsymbol{a}_{q_i}\}_{i=1}^k$
- PFI generalizes as

$$B_k = B_0 E_1 E_2 \ldots E_k$$

- FT generalizes as

$$B_k = L_0 R_1 R_2 \ldots R_k U_k$$

Eventually more computationally efficient or numerically prudent to reinvert some B_k

Simplex method: Schur complement (SC) update for B_k

- After k updates, let

$$\begin{aligned} V_k &= \begin{bmatrix} \boldsymbol{a}_{q_1} & \boldsymbol{a}_{q_2} & \dots & \boldsymbol{a}_{q_k} \end{bmatrix} \\ I_k &= \begin{bmatrix} \boldsymbol{e}_{p_1} & \boldsymbol{e}_{p_2} & \dots & \boldsymbol{e}_{p_k} \end{bmatrix} \end{aligned}$$

- Then

$$B_k = B_0 + (V_k - I_k)I_k^T = B_0[I - (Y_k - I_k)I_k^T] = B_0 H_k$$

where $Y_k = B_0^{-1} V_k$ and

$$H_k^{-1} = I - (Y_k - I_k) C_k^{-1} I_k$$

where $C_k = -I_k Y_k$ is the **Schur complement**

- To operate with C_k^{-1}
    - Bisschop and Meeraus (1977) updated explicit C_k^{-1}
    - Eldersveld and Saunders (1990) updated QR factors of C_k
    - Fletcher and H (1990) updated LU factors of C_k

Novel update techniques for the revised simplex method

Novel update techniques for the revised simplex method: 1

Alternative product form update (APF)

- **Recall:** Column p of B is replaced by $\boldsymbol{a}_q$ to give $\bar{B} = B + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T$
    - Traditional PFI takes B out as a factor on the left so $\bar{B} = BE$
- **Idea:** Why not take it out on the right!

$$\begin{aligned} \bar{B} &= [I + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T B^{-1}]B = TB \\ \text{where} \quad T &= I + (\boldsymbol{a}_q - \boldsymbol{a}_{p'})\widehat{\boldsymbol{e}}_p^T \end{aligned}$$

- T is formed of known data and readily invertible (like E for PFI)
  Naturally compute $\widehat{\boldsymbol{e}}_p$ when solving $B^T\boldsymbol{\pi}_p = \boldsymbol{e}_p$
- **But:** Is this useful?

Novel update techniques for the revised simplex method: 2

Middle product form update (MPF)

- **Recall:** Column p of B is replaced by $\boldsymbol{a}_q$ to give $\bar{B} = B + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T$
- **Idea:** Substitute $B = LU$ and take factors L on the left and U on the right!

$$\begin{aligned}
\bar{B} &= LU + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T \\
&= LU + LL^{-1}(\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T U^{-1} U \\
&= L[I + (\widetilde{\boldsymbol{a}}_q - U\boldsymbol{e}_p)\widetilde{\boldsymbol{e}}_p^T]U \\
&= LTU \qquad \text{where} \quad T = I + (\widetilde{\boldsymbol{a}}_q - \boldsymbol{u}_p)\widetilde{\boldsymbol{e}}_p^T
\end{aligned}$$

- T is formed of known data and readily invertible (like E for PFI)
  Naturally compute $\widetilde{\boldsymbol{a}}_q$ when solving $B\,\widehat{\boldsymbol{a}}_q = \boldsymbol{a}_q$ and $\widetilde{\boldsymbol{e}}_p$ when solving $B^T\boldsymbol{\pi}_p = \boldsymbol{e}_p$
- **But:** Is this useful?

Novel update techniques for the revised simplex method: 3

Forrest-Tomlin update

- **Recall:** Column p of B is replaced by $\boldsymbol{a}_q$ to give $\bar{B} = B + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T$
- **Idea:** Substitute $B = LU$ and take factor L on the left

$$\bar{B} \quad = \quad L[U + (L^{-1}\boldsymbol{a}_q - U\boldsymbol{e}_p)\boldsymbol{e}_p^T] \quad = \quad LU'$$

where $U' = U + (\tilde{\boldsymbol{a}}_q - \boldsymbol{u}_p)\boldsymbol{e}_p^T$ is "spiked" upper triangular and then eliminate

Collective Forrest-Tomlin (CFT) update

- Update Forrest-Tomlin representation of B after multiple basis changes
- Don't have data to perform a sequence of standard FT updates
- Have to perform elimination corresponding to multiple spikes
- **But:** Is this useful?

Huangfu and H (2013)

Novel update techniques for the revised simplex method: Results

- Test environment
  - 30 representative LP problems from standard test sets
  - Same sequence of basis changes for all update techniques
- Geometric mean time for operations to form and use update data

Update	PFI	FT	APF	MPF	CFT
Mean	1.00	0.30	0.95	0.45	0.29

Conclusions:

- FT much better than PFI
- APF little better than PFI
- MPF closer to FT than PFI
- CFT as good as FT

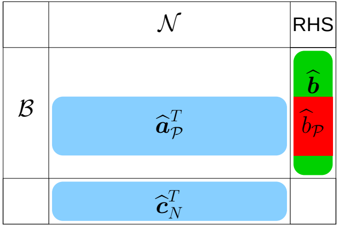
Application: $\mathtt{h_2gmp}$

- Features
  - Model management: Add/delete/modify problem data
  - Functionality: Presolve + crash + advanced basis start
  - Algorithm
    - Dual simplex
    - Steepest edge pricing
    - Bound-flipping ratio test
    - Forrest-Tomlin update for $B := B + (\boldsymbol{a}_q - B\boldsymbol{e}_p)\boldsymbol{e}_p^T$
  - Efficiency: High performance serial and parallel computational components
- Fast
- Reliable
- Open-source C++

H, Huangfu and Galabova (2013–date)

Application: Multiple iteration parallelism with `pami` option in h$_2$gmp

- 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 $\boldsymbol{\pi}_{\mathcal{P}} = B^{-1}\boldsymbol{e}_{\mathcal{P}}$
- 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)

Novel update procedures: are they useful?

Update 1: Alternative product form update (APF)

- **Recall:** $\bar{B} = TB$ with T easily invertible
- Used to get $\boldsymbol{\pi}_{\mathcal{P}} = \bar{B}^{-1}\boldsymbol{e}_{\mathcal{P}}$ from $B^{-1}\boldsymbol{e}_{\mathcal{P}}$ in `pami`
- Used to compute multiple $B^{-1}\boldsymbol{a}_F$ efficiently after multiple BFRT in `pami`

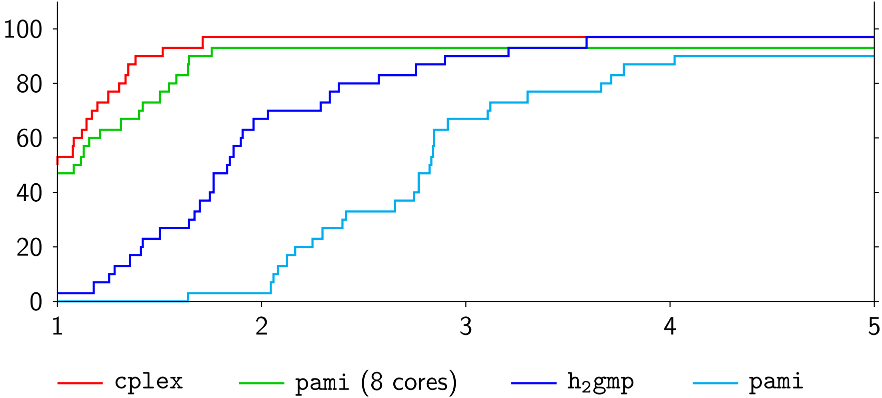
Update 2: Middle product form update (MPF)

- **Recall:** $\bar{B} = LTU$ with T easily invertible
- Not used by `h2gmp`!
- Used by Google in `glop`

Update 3: Collective Forrest-Tomlin update (CFT)

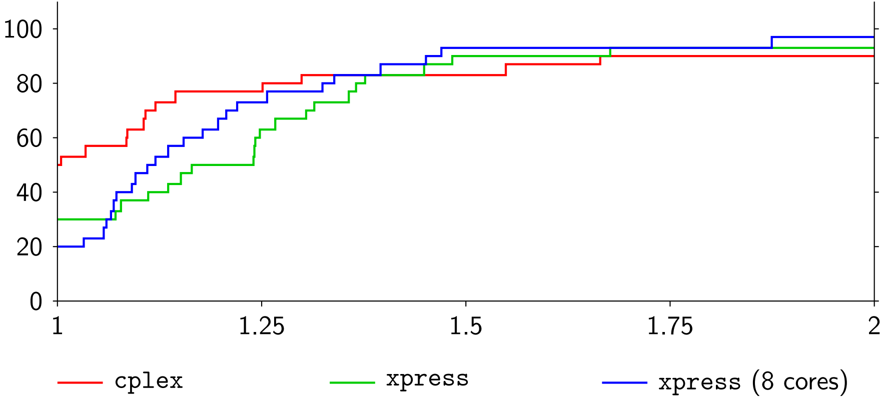
Used to perform multiple Forrest-Tomlin updates after minor iterations in `pami`

Multiple iteration parallelism: cplex vs pami vs $h_2$gmp



- pami is less efficient than $h_2$gmp in serial
- pami speedup more than compensates
- pami performance approaching cplex

Multiple iteration parallelism: cplex vs xpress



- pami ideas incorporated in FICO Xpress (Huangfu 2014)
- Xpress simplex solver now fastest commercial simplex solver

Conclusions

- I've been updating for 30 years: time to reinvert!
- Three novel updates
  - Two very simple: APF and MPF
  - One rather more complex: CFT
- Key components in $\mathtt{h_2gmp}$'s parallel simplex component `pami`

Slides: http://www.maths.ed.ac.uk/hall/EUROPT17/

R. Fletcher and J. A. J. Hall.
Towards reliable linear programming.
In *Pitman Research Notes in Mathematics Series 228*, pages 89–104. Longman Scientific and Technical, 1990.

Q. Huangfu and J. A. J. Hall.
Parallelizing the dual revised simplex method.
Technical Report ERGO-14-011, School of Mathematics, University of Edinburgh, 2014.
Accepted for publication in Mathematical Programming Computation.

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