

High performance solution of linear optimization problems 1

Serial techniques for the simplex method

Julian Hall

School of Mathematics
University of Edinburgh

jajhall@ed.ac.uk

ALOP Autumn School, Trier

31 August 2017



THE UNIVERSITY
of EDINBURGH

High performance solution of linear optimization problems

- More to data science than machine learning and data analysis
- More to optimization than first order methods
- Other application areas yield data science problems intractable using “modern” methods

Need for high performance implementations of “traditional” methods



Weatherford[®]



- Introduction
- LP background
 - Analysis
 - Optimality
- Primal revised simplex method
 - Representing B^{-1} (INVERT)
 - Solving linear systems (FTRAN and BTRAN) with focus on exploiting hyper-sparsity
 - Matrix-vector product $\pi_p^T N$ (PRICE)
- Dual revised simplex method
 - Optimality conditions
 - Dual revised simplex method
 - Dual of a tightened primal problem
 - Dual vs primal simplex method

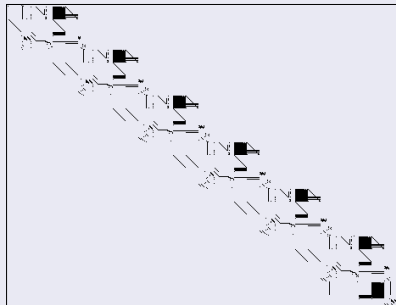
High performance solution of linear programming (LP) problems

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

Background

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

Example



STAIR: 356 rows, 467 columns and 3856 nonzeros

Solving LP problems: The feasible region and LP outcomes

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

The feasible region

- Any point $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ is **feasible**
- The set of all feasible points is the **feasible region** K

Possible outcomes

Exactly one of the following hold for any LP problem

- The feasible region K is empty so the LP is **infeasible**
- The objective may be decreased without bound on K so the LP is **unbounded**
- There is a point $\mathbf{x} \in K$ which is an **optimal solution** of the LP

Solving LP problems: Necessary and sufficient conditions for optimality

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

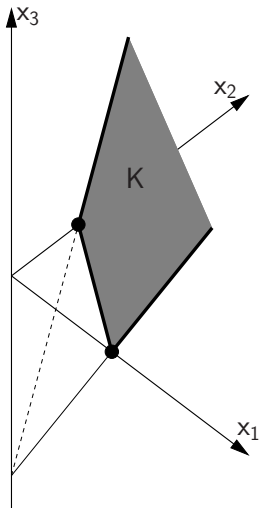
Karush-Kuhn-Tucker (KKT) conditions

$\mathbf{x}^*$ is an optimal solution $\iff$ there exist $\mathbf{y}^*$ and $\mathbf{s}^*$ such that

$$\begin{array}{lll} \mathbf{Ax} = \mathbf{b} & (1) & \mathbf{x} \geq \mathbf{0} \quad (3) \\ A^T \mathbf{y} + \mathbf{s} = \mathbf{c} & (2) & \mathbf{s} \geq \mathbf{0} \quad (4) \end{array} \quad \mathbf{x}^T \mathbf{s} = 0 \quad (5)$$

- For the **simplex algorithm** (1–2 and 5) always hold
 - **Primal** simplex algorithm: (3) holds and the algorithm seeks to satisfy (4)
 - **Dual** simplex algorithm: (4) holds and the algorithm seeks to satisfy (3)
- For **interior point methods** (1–4) hold and the method seeks to satisfy (5)

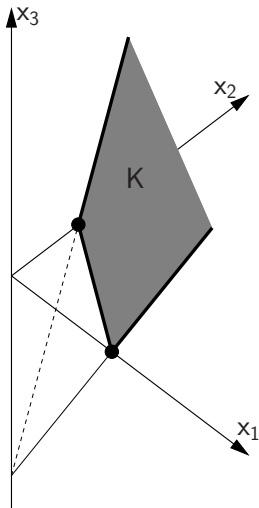
Solving LP problems: Characterizing the feasible region



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

- $A \in \mathbb{R}^{m \times n}$ is full rank
- Solution of $\mathbf{Ax} = \mathbf{b}$ is a $n - m$ dim. **hyperplane** in $\mathbb{R}^n$
- Intersection with $\mathbf{x} \geq \mathbf{0}$ is the **feasible region** K
- A vertex of K has
 - m **basic** components, $i \in \mathcal{B}$ given by $\mathbf{Ax} = \mathbf{b}$
 - $n - m$ zero **nonbasic** components, $j \in \mathcal{N}$where $\mathcal{B} \cup \mathcal{N}$ partitions $\{1, \dots, n\}$
- A solution of the LP occurs at a **vertex** of K

Solving LP problems: Characterizing the feasible region



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

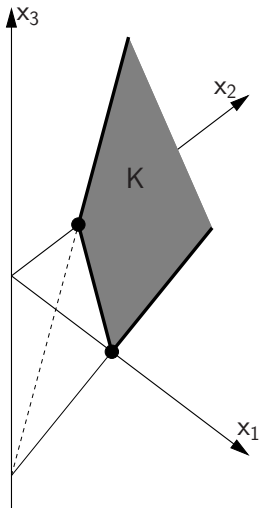
- Equations partitioned according to $\mathcal{B} \cup \mathcal{N}$ as

$$\mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b}$$

with nonsingular **basis matrix** $\mathbf{B}$

- Solution set characterized by $\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{Nx}_N)$
- At a **vertex** of K , $\begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}} \\ \mathbf{0} \end{bmatrix}$ where $\hat{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$
- K is $\mathbf{x}_B = \hat{\mathbf{b}} - \mathbf{B}^{-1}\mathbf{Nx}_N$ for some $\mathbf{x}_N \geq \mathbf{0}$

Solving LP problems: Sufficient conditions for optimality



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

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

$$\begin{aligned} f &= \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ &= \mathbf{c}_B^T (\hat{\mathbf{b}} - B^{-1} N \mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N \\ &= \hat{f} + \hat{\mathbf{c}}_N^T \mathbf{x}_N \end{aligned}$$

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

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

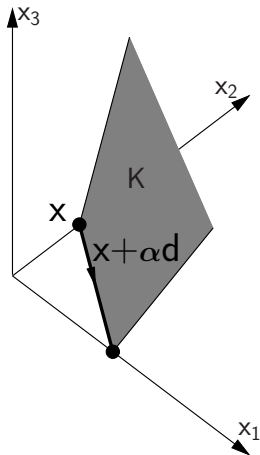
Primal revised simplex method

The simplex algorithm: Why study it



- Old technology?
- Underpins solution methods for MIP and NLP
- Education in high performance numerical linear algebra

The simplex algorithm: Definition



At a feasible vertex $\mathbf{x} = \begin{bmatrix} \hat{\mathbf{b}} \\ 0 \end{bmatrix}$ corresponding to $\mathcal{B} \cup \mathcal{N}$

- ① If $\hat{\mathbf{c}}_{\mathcal{N}} \geq \mathbf{0}$ then **stop: the solution is optimal**
- ② Scan $\hat{c}_j < 0$ for q to leave $\mathcal{N}$
- ③ Let $\hat{\mathbf{a}}_q = B^{-1}N\mathbf{e}_q$ and $\mathbf{d} = \begin{bmatrix} -\hat{\mathbf{a}}_q \\ \mathbf{e}_q \end{bmatrix}$
- ④ Scan $\hat{b}_i / \hat{a}_{iq} > 0$ for α and p to leave $\mathcal{B}$
- ⑤ Exchange p and q between $\mathcal{B}$ and $\mathcal{N}$
- ⑥ Go to 1

The simplex algorithm: Choosing a column

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

Primal algorithm: Assume $\hat{\mathbf{b}} \geq \mathbf{0}$ Seek $\hat{\mathbf{c}}_N \geq \mathbf{0}$

Scan $\hat{c}_j < 0, i \in \mathcal{N}$, for a good candidate q to leave $\mathcal{N}$ CHUZC

Scan $\hat{b}_i / \hat{a}_{iq} > 0, i \in \mathcal{B}$, for a good candidate p to leave $\mathcal{B}$ CHUZR

The simplex algorithm: Choosing a row

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\hat{\mathbf{a}}_q$ $\hat{\mathbf{a}}_{pq}$	$\hat{\mathbf{b}}$ $\hat{b}_p$

Primal algorithm: Assume $\hat{\mathbf{b}} \geq \mathbf{0}$ Seek $\hat{\mathbf{c}}_N \geq \mathbf{0}$

Scan $\hat{c}_j < 0, i \in \mathcal{N}$, for a good candidate q to leave $\mathcal{N}$ CHUZC

Scan $\hat{b}_i / \hat{a}_{iq} > 0, i \in \mathcal{B}$, for a good candidate p to leave $\mathcal{B}$ CHUZR

The simplex algorithm: Update

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\hat{\mathbf{a}}_q$	$\hat{\mathbf{b}}$
	$\hat{a}_{pq}$ $\hat{\mathbf{a}}_p^T$	$\hat{b}_p$
	$\hat{\mathbf{c}}_q$ $\hat{\mathbf{c}}_N^T$	

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

Update $\hat{\mathbf{b}} := \hat{\mathbf{b}} - \alpha_P \hat{\mathbf{a}}_q$ $\alpha_P = \hat{b}_p / \hat{a}_{pq}$ **UPDATE-PRIMAL**

Update $\hat{\mathbf{c}}_N^T := \hat{\mathbf{c}}_N^T + \alpha_D \hat{\mathbf{a}}_p^T$ $\alpha_D = -\hat{c}_q / \hat{a}_{pq}$ **UPDATE-DUAL**

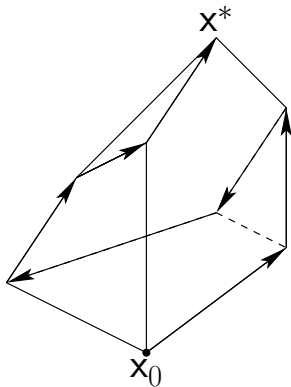
Data required

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

Why does it work?

Objective improves by $\left| \frac{\hat{b}_p \times \hat{c}_q}{\hat{a}_{pq}} \right|$ each iteration

The simplex algorithm: Convergence and complexity



Proof of termination

- In each iteration the objective improves by $\left| \frac{\hat{b}_p \times \hat{c}_q}{\hat{a}_{pq}} \right|$
So, cannot re-visit vertices
- Number of vertices is finite
So, the algorithm terminates at an optimal vertex x^*

Complexity theory

- Might the simplex algorithm visit all $\frac{n!}{m!(n-m)!}$ vertices?
- What is the worst case?

The simplex algorithm: Convergence and complexity

Klee-Minty problems (1971)

$$\begin{aligned} &\text{maximize} && \sum_{j=1}^n 10^{j-1} x_j \\ &\text{subject to} && x_i + 2 \sum_{j=i+1}^n 10^{j-i} x_j \leq 100^{n-i} \quad \text{for } i = 1, \dots, n \\ &&& x_1 \geq 0, \dots, x_n \geq 0 \end{aligned}$$

- Problems have n variables and n constraints
- Feasible region has 2^n vertices
- Simplex algorithm visits all of them! It is *exponential* in time

In practice the simplex algorithm visits $O(m+n) \ll \frac{n!}{m!(n-m)!}$ vertices

Simplex method: Computation

Standard simplex method (SSM): Major computational component

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\hat{N}$	$\hat{\mathbf{b}}$
	$\hat{\mathbf{c}}_N^T$	

Update of tableau: $\hat{N} := \hat{N} - \frac{1}{\hat{a}_{pq}} \hat{\mathbf{a}}_q \hat{\mathbf{a}}_p^T$

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

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

Revised simplex method (RSM): Major computational components

Pivotal row via $B^T \pi_p = \mathbf{e}_p$ **BTRAN** and $\hat{\mathbf{a}}_p^T = \pi_p^T N$ **PRICE**

Pivotal column via $B \hat{\mathbf{a}}_q = \mathbf{a}_q$ **FTRAN** Represent B^{-1} **INVERT**

Simplex method: Representing B^{-1} (INVERT)

- **Aim:** Find representation of B^{-1}
- Determine row and column permutations P and Q so the decomposition

$$PBQ = LU$$

has low fill-in

- P and Q correspond to Gaussian elimination **pivot sequence** $\{(i_k, j_k)\}_{k=1}^m$
- Finding P and Q for minimum global fill-in is NP-hard: Use local heuristic

Simplex method: Representing B^{-1} (INVERT)

- In stage k of Gaussian elimination:

$$P_k B Q_k = \begin{bmatrix} L_{1k} & \\ L_{2k} & I \end{bmatrix} \begin{bmatrix} U_{1k} & U_{2k} \\ & B_k \end{bmatrix}$$

- Pivot selection based on **Markowitz merit**

$$m_{ij} = (r_i - 1)(c_j - 1)$$

where

- r_i is the number of nonzeros in row i of B_k
- c_j is the number of nonzeros in column j of B_k
- Choose (i_k, j_k) over nonzeros in B_k so that
 - $m_{i_k j_k}$ is minimal
 - $|b_{i_k j_k}| > u \max_i |b_{i j_k}|$ where u is 0.1, 0.01, or less!
- For early/many/all k , $m_{i_k j_k} = 0$

Simplex method: Representing B^{-1} (INVERT)

Given: $PBQ = LU$

- P and Q can be accumulated into B so assume $P = Q = I$
- Solve $B\mathbf{x} = \mathbf{b}$ as
 - ① $L\mathbf{y} = \mathbf{b}$ using `ftranL(L,  $\mathbf{b}$ ,  $\mathbf{y}$ )`
 - ② $U\mathbf{x} = \mathbf{y}$ using `ftranU(U,  $\mathbf{y}$ ,  $\mathbf{x}$ )`
- For L and U stored column-wise, use

```
function ftranL( $L$ ,  $\mathbf{b}$ ,  $\mathbf{x}$ )  
   $\mathbf{r} = \mathbf{b}$   
  for all  $j \in \{1, \dots, m\}$  do  
    for all  $i : L_{ij} \neq 0$  do  
       $r_i = r_i - L_{ij}r_j$   
   $\mathbf{x} = \mathbf{r}$ 
```

```
function ftranU( $U$ ,  $\mathbf{b}$ ,  $\mathbf{x}$ )  
   $\mathbf{r} = \mathbf{b}$   
  for all  $j \in \{m, \dots, 1\}$  do  
     $r_j = r_j / U_{jj}$   
    for all  $i : U_{ij} \neq 0$  do  
       $r_i = r_i - U_{ij}r_j$   
   $\mathbf{x} = \mathbf{r}$ 
```

Simplex method: Hyper-sparsity

Recall: major computational components

- **BTRAN**: Form $\pi_p = B^{-T} \mathbf{e}_p$
- **PRICE**: Form $\hat{\mathbf{a}}_p^T = \pi_p^T N$
- **FTRAN**: Form $\hat{\mathbf{a}}_q = B^{-1} \mathbf{a}_q$

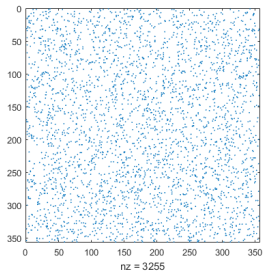
Phenomenon of hyper-sparsity

- Vectors $\mathbf{e}_p$ and $\mathbf{a}_q$ are sparse
- Results π_p , $\hat{\mathbf{a}}_p^T$ and $\hat{\mathbf{a}}_q$ may be sparse—because B^{-1} is sparse
 - In **BTRAN**, π_p is a row of B^{-1}
 - In **PRICE**, $\hat{\mathbf{a}}_p^T$ is a linear combination of a few rows of N
 - In **FTRAN**, $\hat{\mathbf{a}}_q$ is a linear combination of a few rows of B^{-1}

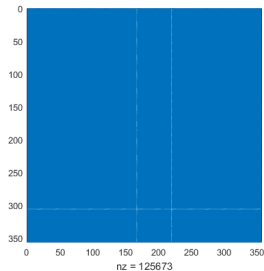
Simplex method: Hyper-sparsity

Simplex method: Inverse of a sparse matrix and solution of $Bx = b$

Random sparse matrix B
 $m = 356$ and density 2.5%



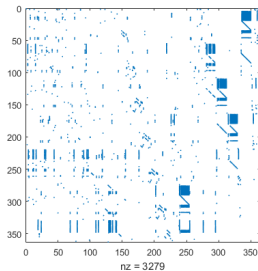
B^{-1} has density of 99%



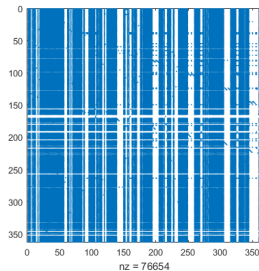
Simplex method: Hyper-sparsity

Simplex method: Inverse of a sparse matrix and solution of $Bx = b$

Optimal B for LP problem stair
 $m = 356$ and density 2.5%



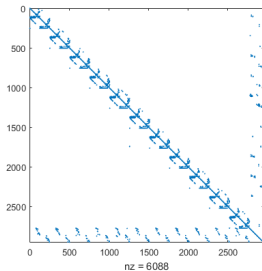
B^{-1} has density of 58%
 $B^{-1}b$ for sparse b is typically dense



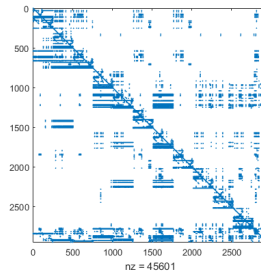
Simplex method: Hyper-sparsity

Simplex method: Inverse of a sparse matrix and solution of $Bx = b$

Optimal B for LP problem pds-02
 $m = 2953$ and density 0.07%



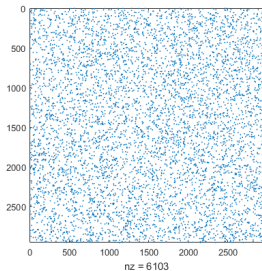
B^{-1} has density of 0.52%
 $B^{-1}b$ for sparse b is typically **sparse**



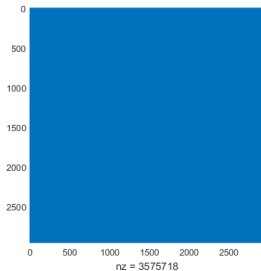
Simplex method: Hyper-sparsity

Simplex method: Inverse of a sparse matrix and solution of $Bx = b$

Random matrix
 $m = 2953$ and density 0.07%



B^{-1} has density of 100%



Simplex method: Hyper-sparsity

- Use solution of $L\mathbf{x} = \mathbf{b}$
 - To illustrate the phenomenon of hyper-sparsity
 - To demonstrate how to exploit hyper-sparsity
- Apply principles to other triangular solves in the simplex method

Simplex method: Hyper-sparsity

Recall: Solve $L\mathbf{x} = \mathbf{b}$ using

```
function ftranL( $L, \mathbf{b}, \mathbf{x}$ )  
   $\mathbf{r} = \mathbf{b}$   
  for all  $j \in \{1, \dots, m\}$  do  
    for all  $i : L_{ij} \neq 0$  do  
       $r_i = r_i - L_{ij}r_j$   
   $\mathbf{x} = \mathbf{r}$ 
```

When $\mathbf{b}$ is **sparse**

- Inefficient until $\mathbf{r}$ fills in

Simplex method: Hyper-sparsity

Better: Check r_j for zero

```
function ftranL( $L, \mathbf{b}, \mathbf{x}$ )  
   $\mathbf{r} = \mathbf{b}$   
  for all  $j \in \{1, \dots, m\}$  do  
    if  $r_j \neq 0$  then  
      for all  $i : L_{ij} \neq 0$  do  
         $r_i = r_i - L_{ij}r_j$   
  
   $\mathbf{x} = \mathbf{r}$ 
```

When $\mathbf{x}$ is **sparse**

- Few values of r_j are nonzero
- Check for zero dominates
- Requires more efficient identification of set $\mathcal{X}$ of indices j such that $r_j \neq 0$

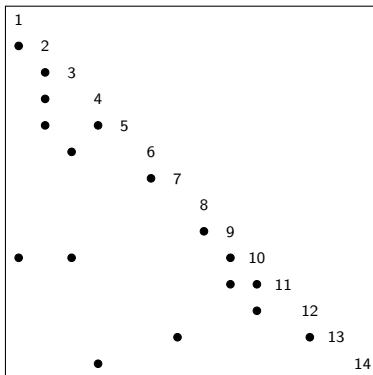
Gilbert and Peierls (1988)

H and McKinnon (1998–2005)

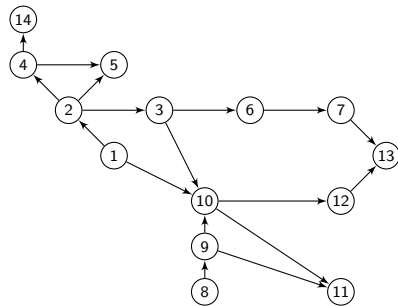
COAP best paper prize (2005)

Exploiting hyper-sparsity in $Lx = b$

Nonzero pattern of L

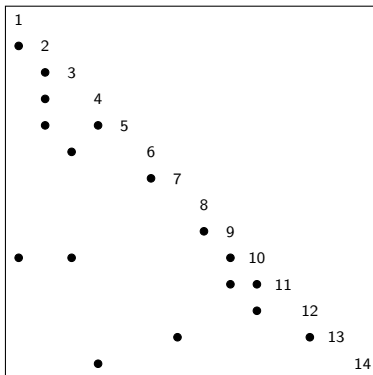


Directed graph G_L

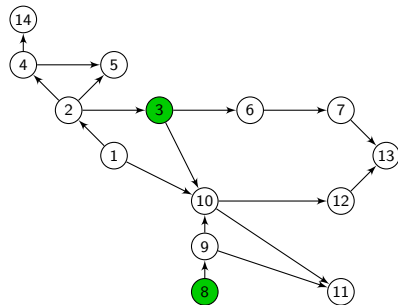


Exploiting hyper-sparsity in $Lx = b$

Nonzero pattern of L



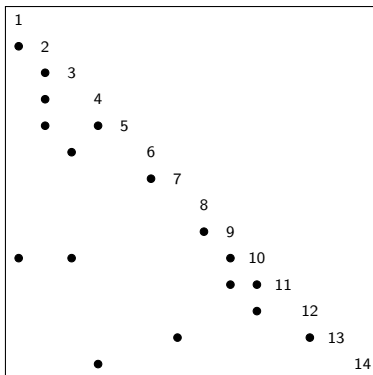
Directed graph G_L



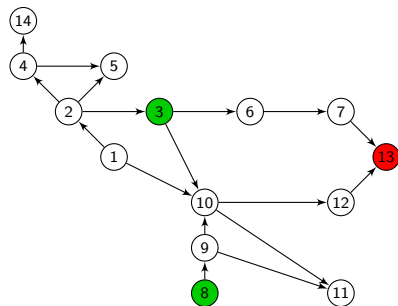
Find $\mathcal{X}$ for $b_3 \neq 0$ and $b_8 \neq 0$

Exploiting hyper-sparsity in $Lx = b$

Nonzero pattern of L



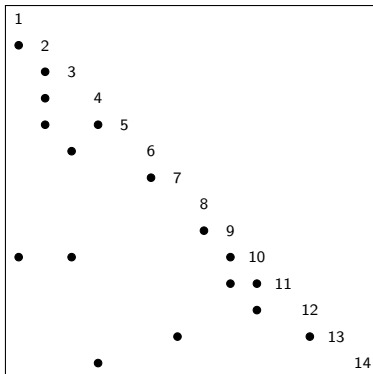
Directed graph G_L



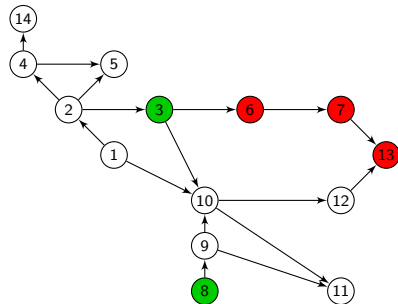
$$\mathcal{X} = \{13\}$$

Exploiting hyper-sparsity in $Lx = b$

Nonzero pattern of L



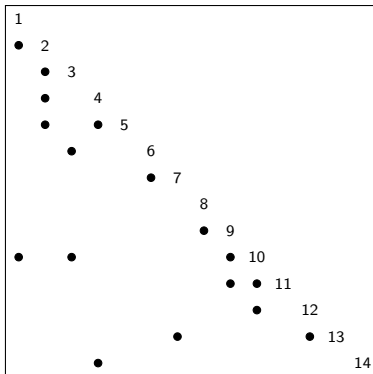
Directed graph G_L



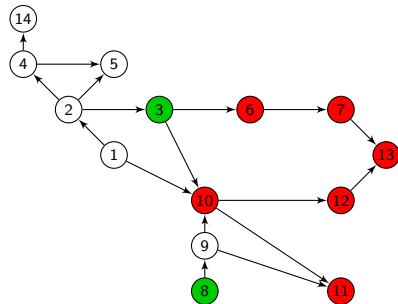
$$\mathcal{X} = \{6, 7, 13\}$$

Exploiting hyper-sparsity in $Lx = b$

Nonzero pattern of L



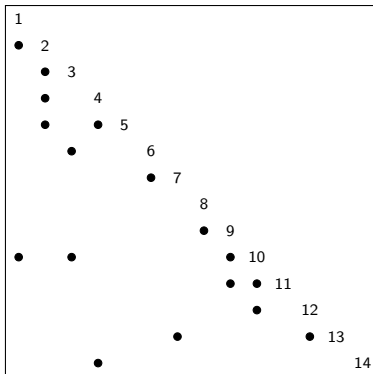
Directed graph G_L



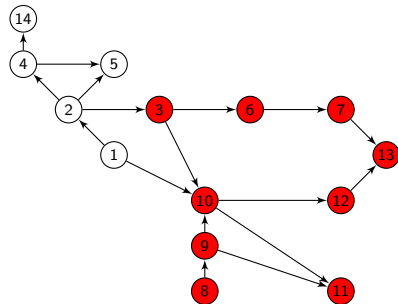
$$\mathcal{X} = \{12, 11, 6, 7, 13\}$$

Exploiting hyper-sparsity in $Lx = b$

Nonzero pattern of L



Directed graph G_L



$$\mathcal{X} = \{8, 9, 3, 10, 12, 11, 6, 7, 13\}$$

Exploiting hyper-sparsity in $Lx = b$

```
function  $\mathcal{X} = \text{reach}(L, \mathbf{b})$   
    assume all nodes are unmarked  
    for all  $i$  for which  $b_i \neq 0$  do  
        if node  $i$  is unmarked then  
             $\text{dfs}(i)$ 
```

```
function  $\text{dfs}(j)$   
    mark node  $j$   
    for all  $i$  for which  $L_{ij} \neq 0$  do  
        if node  $i$  is unmarked then  
             $\text{dfs}(i)$   
    push  $j$  onto stack for  $\mathcal{X}$ 
```

```
function  $\text{ftranL}(L, \mathbf{b}, \mathbf{x})$   
    compute  $\mathcal{X} = \text{reach}(L, \mathbf{b})$   
     $\mathbf{r} = \mathbf{b}$   
    for all  $j \in \mathcal{X}$  do  
        for all  $i$  for which  $L_{ij} \neq 0$  do  
             $r_i = r_i - L_{ij}r_j$   
     $\mathbf{x} = \mathbf{r}$ 
```

Davis (2006)

Simplex method: FTRAN following INVERT

Recall: Given $B = LU$, solve $B\mathbf{x} = \mathbf{b}$ as

- ① $L\mathbf{y} = \mathbf{b}$ using `ftranL(L,  $\mathbf{b}$ ,  $\mathbf{y}$ )`
- ② $U\mathbf{x} = \mathbf{y}$ using `ftranU(U,  $\mathbf{y}$ ,  $\mathbf{x}$ )`

Simplex method: BTRAN following INVERT

- Given $B = LU$, solve $B^T \mathbf{x} = \mathbf{b}$ as
 - 1 $U^T \mathbf{y} = \mathbf{b}$ using `btranU( $U, \mathbf{b}, \mathbf{y}$ )`
 - 2 $L^T \mathbf{x} = \mathbf{y}$ using `btranL( $L, \mathbf{y}, \mathbf{x}$ )`
- For L and U stored column-wise, use

```
function btranU( $U, \mathbf{b}, \mathbf{x}$ )  
   $\mathbf{r} = \mathbf{b}$   
  for all  $j \in \{1, \dots, m\}$  do  
     $\mu = 0$   
    for all  $i : U_{ij} \neq 0$  do  
       $\mu = \mu - U_{ij} r_i$   
       $r_j = (r_j + \mu) / U_{jj}$   
   $\mathbf{x} = \mathbf{r}$ 
```

```
function btranL( $L, \mathbf{b}, \mathbf{x}$ )  
   $\mathbf{r} = \mathbf{b}$   
  for all  $j \in \{m, \dots, 1\}$  do  
     $\mu = 0$   
    for all  $i : L_{ij} \neq 0$  do  
       $\mu = \mu - L_{ij} r_i$   
       $r_j = r_j + \mu$   
   $\mathbf{x} = \mathbf{r}$ 
```

Problem: Cannot identify $\mu = 0$ without computing it!

Solution: Store L and U row-wise as well as column-wise!

Simplex method: Matrix-vector product $\pi_p^T N$ (PRICE)

- Given $\pi_p = B^{-T} \mathbf{e}_p$, form $\hat{\mathbf{a}}_p^T = \pi_p^T N$
- N is naturally stored column-wise, so form $\hat{a}_{pj} = \pi_p^T \mathbf{a}_j$ for $j \in \mathcal{N}$
- **Problems**
 - When π_p^T is sparse there are many operations with zero
 - Particularly inefficient when $\hat{\mathbf{a}}_p^T$ is sparse

- **Solution:** Also store N **row-wise** as $N = \begin{bmatrix} \mathbf{n}_1^T \\ \vdots \\ \mathbf{n}_m^T \end{bmatrix}$

- Form $\hat{\mathbf{a}}_p^T = \pi_p^T \begin{bmatrix} \mathbf{n}_1^T \\ \vdots \\ \mathbf{n}_m^T \end{bmatrix} = \sum_{\pi_{p_i} \neq 0} \pi_{p_i} \mathbf{n}_i^T$

Dual revised simplex method

Solving LP problems: Duality

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

The dual LP problem

$$\text{maximize } f_D = \mathbf{b}^T \mathbf{y} \quad \text{subject to } A^T \mathbf{y} + \mathbf{s} = \mathbf{c} \quad \mathbf{s} \geq \mathbf{0}$$

- Elegant mathematical properties
- Valuable practical properties

Solving dual LP problems: Characterizing the feasible region

- Consider the **dual problem**

$$\text{maximize } f_D = \mathbf{b}^T \mathbf{y} \quad \text{subject to } A^T \mathbf{y} + \mathbf{s} = \mathbf{c} \quad \mathbf{s} \geq \mathbf{0}$$

- For a partition $\mathcal{B} \cup \mathcal{N}$ of the equations with nonsingular **basis matrix** B

$$\begin{bmatrix} B^T \\ N^T \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{s}_B \\ \mathbf{s}_N \end{bmatrix} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix}$$

- So $\mathbf{y} = B^{-T}(\mathbf{c}_B - \mathbf{s}_B)$ and $\mathbf{s}_N = \mathbf{c}_N - N^T \mathbf{y} = \hat{\mathbf{c}}_N + N^T B^{-T} \mathbf{s}_B$

- At a **vertex** of the dual feasible region K_D , $\begin{bmatrix} \mathbf{y} \\ \mathbf{s}_N \\ \mathbf{s}_B \end{bmatrix} = \begin{bmatrix} B^{-T} \mathbf{c}_B \\ \hat{\mathbf{c}}_N \\ \mathbf{0} \end{bmatrix}$, where $\hat{\mathbf{c}}_N \geq \mathbf{0}$

- Points in K_D are given by $\mathbf{s}_B \geq \mathbf{0}$

Solving dual LP problems: Optimality conditions

- Consider the **dual problem**

$$\text{maximize } f_D = \mathbf{b}^T \mathbf{y} \quad \text{subject to } A^T \mathbf{y} + \mathbf{s} = \mathbf{c} \quad \mathbf{s} \geq \mathbf{0}$$

- Substitute $\mathbf{y} = B^{-T}(\mathbf{c}_B - \mathbf{s}_B)$ into the objective to give

$$f_D = \mathbf{b}^T B^{-T}(\mathbf{c}_B - \mathbf{s}_B) = \hat{f} - \hat{\mathbf{b}}^T \mathbf{s}_B$$

- Partition yields an optimal solution if there is
 - Dual feasibility $\hat{\mathbf{c}}_N \geq \mathbf{0}$
 - Primal feasibility $\hat{\mathbf{b}} \geq \mathbf{0}$
- Partition is optimal for primal $\iff$ partition is optimal for dual
- Dual simplex algorithm** for an LP is *primal algorithm* applied to the *dual problem*
- Structure of dual equations allows dual simplex algorithm to be applied to primal simplex tableau

Dual simplex algorithm

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

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

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

	$\mathcal{N}$	RHS
$\mathcal{B}$		$\hat{\mathbf{b}}$ $\hat{b}_p$

Dual simplex algorithm

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

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

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

Dual step

- Increase s_p from zero
- Maintain dual feasibility $\mathbf{s}_N \geq \mathbf{0}$

$$\iff \hat{\mathbf{c}}_N + N^T B^{-T} \mathbf{e}_p s_p \geq \mathbf{0}$$

$$\iff \hat{\mathbf{c}}_N^T + s_p \hat{\mathbf{a}}_p^T \geq \mathbf{0}$$

$$\iff s_p \leq \min_{\hat{a}_{pj} < 0} -\frac{\hat{c}_j}{\hat{a}_{pj}}$$

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\hat{a}_{pq}$ $\hat{\mathbf{a}}_p^T$	
	$\hat{c}_q$ $\hat{\mathbf{c}}_N^T$	

Dual simplex algorithm

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

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

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

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

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

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

	$\mathcal{N}$		RHS
$\mathcal{B}$	$\hat{\mathbf{a}}_q$		$\hat{\mathbf{b}}$
	$\hat{a}_{pq}$	$\hat{\mathbf{a}}_p^T$	$\hat{b}_p$
	$\hat{c}_q$	$\hat{\mathbf{c}}_N^T$	

Data required

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

Why does it work?

Objective improves by $\frac{\hat{b}_p \times \hat{c}_q}{\hat{a}_{pq}}$ each iteration

Simplex method: Computation

Standard simplex method (SSM): Major computational component

	$\mathcal{N}$	RHS
$\mathcal{B}$	$\hat{N}$	$\hat{\mathbf{b}}$
	$\hat{\mathbf{c}}_N^T$	

Update of tableau: $\hat{N} := \hat{N} - \frac{1}{\hat{a}_{pq}} \hat{\mathbf{a}}_q \hat{\mathbf{a}}_p^T$

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

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

Revised simplex method (RSM): Major computational components

Pivotal row via $B^T \pi_p = \mathbf{e}_p$ **BTRAN** and $\hat{\mathbf{a}}_p^T = \pi_p^T N$ **PRICE**

Pivotal column via $B \hat{\mathbf{a}}_q = \mathbf{a}_q$ **FTRAN** Represent B^{-1} **INVERT**

LP duality: Adding a constraint to the primal problem

- For the primal LP

$$\text{minimize } f = \mathbf{c}^T \mathbf{x} \quad \text{subject to } A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \quad (P)$$

consider adding the constraint $\mathbf{a}^T \mathbf{x} = b$ to form the problem

$$\text{minimize } f = \mathbf{c}^T \mathbf{x} \quad \text{subject to } \begin{bmatrix} A \\ \mathbf{a}^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ b \end{bmatrix}, \quad \mathbf{x} \geq \mathbf{0} \quad (P')$$

- This is a **tightening** of (P)
- An optimal solution $\mathbf{x}^*$ of (P) either
 - Satisfies $\mathbf{a}^T \mathbf{x}^* = b$, so is a feasible solution of (P') —and hence solves (P')
 - Does not satisfy $\mathbf{a}^T \mathbf{x}^* = b$, so is not a feasible solution of (P')
So feasibility must be restored to solve (P')

LP duality: Adding a constraint to (P) adds a variable to (D)

- The corresponding dual problem is

$$\text{maximize } f = \mathbf{b}^T \mathbf{y} + by \quad \text{subject to } A^T \mathbf{y} + \mathbf{a}y + \mathbf{s} = \mathbf{c}, \quad \mathbf{s} \geq \mathbf{0} \quad (D')$$

- This is a **relaxation** of (D)
 - Given an optimal solution $\mathbf{y}^*$ for (D)
 - $\mathbf{y}^*$ and $y = 0$ are a feasible solution of (D')
Only have to find an optimal solution of (D')
 - Solving after adding one/more constraints is important in **branch-and-bound** for integer programming
 - Retaining dual feasibility makes it more attractive to solve the dual problem or use the dual simplex method

Simplex algorithm: Primal or dual?

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