

NATCOR Convex Optimization

Linear Programming 1

Julian Hall

School of Mathematics
University of Edinburgh

jajhall@ed.ac.uk

5 June 2018

What is linear programming (LP)?

- The most important model used in **optimal decision-making**
- Grew out of US Army Air Force logistics problems in WW2
- First practical LP problem was formulated by George Dantzig in 1946
- Dantzig invented the principal solution technique—the **simplex algorithm**—in 1947
 - In the “**Top 10 algorithms of the 20th century**”
 - “**The algorithm that runs the world**”
- LP and the simplex method have revolutionised organised human decision-making

New Scientist (2012)



G. B. Dantzig (1914-2005)



What is linear programming (LP)?



The diet problem

- Given
 - Costs of foodstuffs
 - Nutrient information and daily requirements
- What is the cheapest diet?

A linear programming model

- Let x_j be the amount of food j purchased ($x_j \geq 0$), for $j = 1, \dots, n$

- Cost c_j of food j gives total cost $f = \sum_{j=1}^n c_j x_j$ to be minimized

- Let a_{ij} be the amount of nutrient i in food j




Matching the daily requirement b_i gives $\sum_{j=1}^n a_{ij} x_j = b_i$, for $i = 1, \dots, m$

Solving LP problems: The standard simplex method

General LP problem in **standard form** is

$$\text{maximize } f = \mathbf{c}^T \mathbf{x} \quad \text{such that } A\mathbf{x} = \mathbf{b} \quad \text{and } \mathbf{x} \geq \mathbf{0}$$

with m equations and n variables ($m < n$)

	\mathcal{N}	RHS
\mathcal{B}		
		

The standard (tableau) simplex method

- Choose a column q with positive entry in $\hat{\mathbf{c}}$
- Choose a row p with least ratio between components in $\hat{\mathbf{b}}$ and $\hat{\mathbf{a}}_q$
- Update the tableau

Need a better idea of what's going on and how to solve LP problems efficiently

Solving LP problems: The feasible region

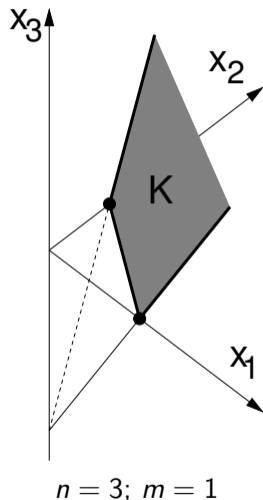
General LP problem is

maximize $f = \mathbf{c}^T \mathbf{x}$ such that $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$

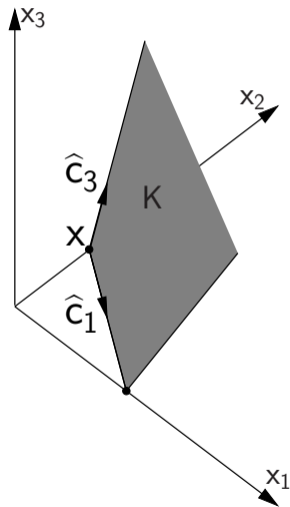
with m equations and n variables ($m < n$)

The feasible region

- Solution of $A\mathbf{x} = \mathbf{b}$ is a $n - m$ dimensional **hyperplane** in \mathbb{R}^n
- Intersection with $\mathbf{x} \geq \mathbf{0}$ is the **feasible region** K
 - K is a **polyhedron**
 - A vertex has
 - $n - m$ zero components
 - m components given by $A\mathbf{x} = \mathbf{b}$
- A solution of the LP occurs at a **vertex** of K



The simplex algorithm: Choosing where to move

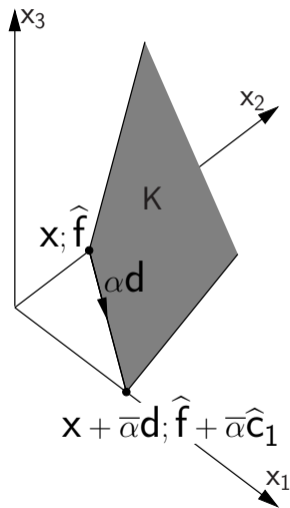


General LP problem is

$$\text{maximize } f = \mathbf{c}^T \mathbf{x} \quad \text{such that} \quad \mathbf{Ax} = \mathbf{b} \quad \text{and} \quad \mathbf{x} \geq \mathbf{0}$$

- At a vertex \mathbf{x} , consider moving along an edge of K to improve the value of f
 - Corresponds to increasing a variable x_j from zero
 - Rate of change \hat{c}_j of f with x_j is known
- If no positive \hat{c}_j then \mathbf{x} is optimal so **stop**
- Increase variable x_q with greatest \hat{c}_j

The simplex algorithm: Choosing how far to go



- Increase variable x_q with greatest $\hat{c}_j > 0$
- Edge direction \mathbf{d} corresponding x_q is known
- Move along $\mathbf{x} + \alpha \mathbf{d}$
- Stop when first component of $\mathbf{x} + \alpha \mathbf{d}$ is zeroed
 x_q increases to $\bar{\alpha} > 0$
- Objective increases by $\bar{\alpha} \hat{c}_q > 0$
- Repeat!

The simplex algorithm: Basic and nonbasic variables

At a vertex \mathbf{x}

- There are $n - m$ **nonbasic variables** \mathbf{x}_N with value zero
 - Their indices form a set \mathcal{N}
 - The columns of A corresponding to \mathcal{N} form the matrix N
 - The components of \mathbf{c} corresponding to \mathcal{N} form the vector \mathbf{c}_N
- There are m **basic variables** \mathbf{x}_B with values uniquely defined by the m equations
 - Their indices form a set \mathcal{B}
 - The columns of A corresponding to \mathcal{B} form the nonsingular **basis matrix** B
 - The components of \mathbf{c} corresponding to \mathcal{B} form the vector \mathbf{c}_B

The **partitioned LP** in standard form is then

$$\begin{aligned} \text{maximize } f &= \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ \text{subject to } & B\mathbf{x}_B + N\mathbf{x}_N = \mathbf{b} \\ & \mathbf{x}_B \geq \mathbf{0}, \mathbf{x}_N \geq \mathbf{0} \end{aligned}$$

The simplex algorithm: The reduced objective and optimality

- Using the partitioned equations

$$B\mathbf{x}_B + N\mathbf{x}_N = \mathbf{b} \quad \Rightarrow \quad \mathbf{x}_B = B^{-1}(\mathbf{b} - N\mathbf{x}_N) = \hat{\mathbf{b}} - B^{-1}N\mathbf{x}_N \quad \text{where} \quad \hat{\mathbf{b}} = B^{-1}\mathbf{b}$$

- Hence, substituting for \mathbf{x}_B , the objective function is

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

where

- $\hat{f} = \mathbf{c}_B^T \hat{\mathbf{b}}$ is the objective value when $\mathbf{x}_N = \mathbf{0}$
- $\hat{\mathbf{c}} = \mathbf{c}_N - N^T B^{-T} \mathbf{c}_B$ is the vector of **reduced costs**
- Behaviour of f as \mathbf{x}_N increases from zero yields the **optimality condition** $\hat{\mathbf{c}} \leq \mathbf{0}$

This is the key to solving LP problems

The simplex algorithm: Data requirements

Each simplex iteration requires

- Reduced costs $\hat{\mathbf{c}} = \mathbf{c}_N - N^T B^{-T} \mathbf{c}_B$
- Reduced RHS $\hat{\mathbf{b}} = B^{-1} \mathbf{b}$
- Edge direction \mathbf{d} corresponding to increasing x_q
 - Know $\mathbf{x}_B = \hat{\mathbf{b}} - B^{-1} N \mathbf{x}_N$
 - $\mathbf{x}_N = x_q \mathbf{e}_q$ when increasing just x_q
 - Hence $\mathbf{x}_B = \hat{\mathbf{b}} - x_q \times B^{-1} N \mathbf{e}_q$

Edge direction is thus

$$\mathbf{d} = \begin{bmatrix} \mathbf{e}_q \\ -\hat{\mathbf{a}}_q \end{bmatrix} \quad \text{where} \quad \hat{\mathbf{a}}_q = B^{-1}(N \mathbf{e}_q) = B^{-1} \mathbf{a}_q$$

	\mathcal{N}	RHS
B		

The simplex algorithm: Computational requirements

- Find reduced costs $\hat{\mathbf{c}} = \mathbf{c}_N - N^T(B^{-T}\mathbf{c}_B)$ thus:
 - Solve $B^T\boldsymbol{\pi} = \mathbf{c}_B$
 - Form $\hat{\mathbf{c}} = \mathbf{c}_N - N^T\boldsymbol{\pi}$
- Find basic edge direction $\hat{\mathbf{a}}_q = B^{-1}\mathbf{a}_q$ thus:
 - Solve $B\hat{\mathbf{a}}_q = \mathbf{a}_q$
- Reduced RHS $\hat{\mathbf{b}}$ is maintained by updating

The revised simplex method

Data required by the algorithm can be found by solving *square* systems of equations

$$B^T\boldsymbol{\pi} = \mathbf{c}_B \quad \text{and} \quad B\hat{\mathbf{a}}_q = \mathbf{a}_q$$

Efficiency depends on

- How B^{-1} is represented
- How the following relation between successive matrices B is exploited

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

The simplex algorithm: Efficient implementation

Standard simplex method

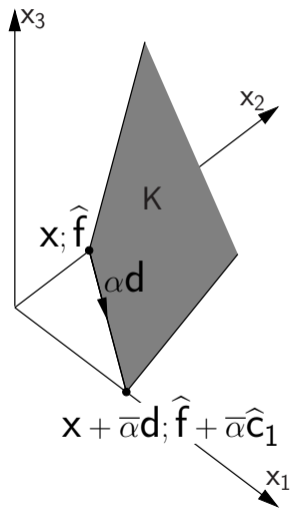
- Maintains $B^{-1}N$, $\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$ in a rectangular tableau
- Requires $O(mn)$ storage and $O(mn)$ computation per iteration
- Inefficient and prohibitively expensive for large problems

Revised simplex method

- Computes $\hat{\mathbf{a}}_q = B^{-1}\mathbf{a}_q$ as required, forms $\hat{\mathbf{c}}$ and updates $\hat{\mathbf{b}} = B^{-1}\mathbf{b}$
- Requires up to $O(m^2)$ storage and $O(m^2)$ computation per iteration but vastly less for sparse LP problems
- Efficient for large (sparse) problems

Important to exploit **matrix sparsity**

The simplex algorithm: Recap



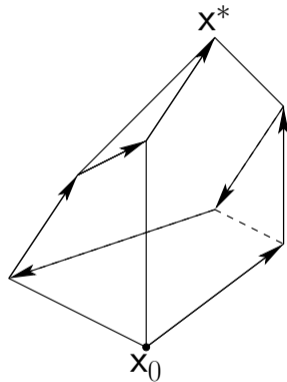
- Increase variable x_q with greatest $\hat{c}_j > 0$
- Edge direction \mathbf{d} corresponding x_q is known
- Move along $\mathbf{x} + \alpha \mathbf{d}$
- Stop when first component of $\mathbf{x} + \alpha \mathbf{d}$ is zeroed
 x_q increases to $\bar{\alpha} > 0$
- Objective increases by $\bar{\alpha} \hat{c}_q > 0$
- Repeat!

The simplex algorithm: Does it terminate? In how many iterations?

Proof of termination

- In each iteration the objective increases by $\bar{\alpha} \hat{c}_q > 0$
So, cannot re-visit vertices
- Number of vertices is finite
So, the algorithm terminates

- In practice the algorithm visits $O(m + n) \ll \frac{n!}{m!(n-m)!}$ vertices
- Could it visit all the vertices?



The simplex algorithm: The worst case *can* happen!

Klee-Minty problems

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

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

Led to better simplex variants which weight \hat{c}_j by $\|\mathbf{d}_j\|$ Goldfarb and Reid (1977)

Bigger LP problems

Suppose N people are choosing diets with limits on the available food

- Person k has an individual LP $\min \mathbf{c}_k^T \mathbf{x}_k$ s.t. $A_k \mathbf{x}_k = \mathbf{b}_k$ and $\mathbf{x}_k \geq \mathbf{0}$
- Person k consumes $x_{kj} = [\mathbf{x}_k]_j$ of food j so $\sum_{k=1}^N x_{kj} \leq b_j$
- This **availability constraint** links the individual LPs

Dantzig-Wolfe structure

$$\begin{array}{ll} \min & f = \mathbf{c}_1^T \mathbf{x}_1 + \dots + \mathbf{c}_N^T \mathbf{x}_N \\ \text{s.t.} & A_{01} \mathbf{x}_1 + \dots + A_{0N} \mathbf{x}_N \leq \mathbf{b}_0 \\ & A_1 \mathbf{x}_1 = \mathbf{b}_1 \\ & \vdots \\ & A_N \mathbf{x}_N = \mathbf{b}_N \\ & \mathbf{x}_1 \geq \mathbf{0} \quad \dots \quad \mathbf{x}_N \geq \mathbf{0} \end{array}$$

Solution methods can exploit problem **structure**

Simplex method: Why use it?

- Interior point methods (IPM) are the “modern” alternative to the simplex method
- For single LP problems IPM are generally faster
- For some classes of single LP problems the simplex method is faster
- When solving sequences of related LP problems the simplex method is preferable
 - Branch-and-bound for discrete optimization
 - Sequential linear programming for nonlinear optimization
- Why?
 - Simplex method yields a basic feasible solution
 - Simplex method can be re-started easily from an optimal solution of one LP to solve a related LP quickly
- 71 years old and going strong!

- Described the simplex algorithm
- Identified that efficient implementation depends on techniques for solving related systems of equations
- Remains to consider how to exploit LP problem **structure and matrix sparsity**:
Wednesday 14:30–15:30; 16:00–17:00