NATCOR Convex Optimization
Linear Programming 1

Julian Hall

School of Mathematics
University of Edinburgh

jajhall@ed.ac.uk

14 June 2016

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"**

New Scientist (2012)

- LP and the simplex method have revolutionised organised human decision-making

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 $j = 1, \ldots, n$
- 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

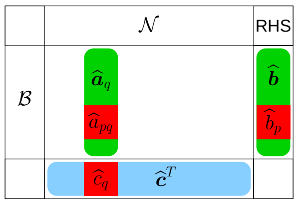
  Matching the daily requirement b_i gives $\sum_{j=1}^{n} a_{ij} x_j = b_i$, for $i = 1, \ldots, m$

Solving LP problems: The standard simplex method

General LP problem in **standard form** is

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

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



The standard (tableau) simplex method

- Choose a column q with positive entry in $\widehat{\boldsymbol{c}}$
- Choose a row p with least ratio between components in $\widehat{\boldsymbol{b}}$ and $\widehat{\boldsymbol{a}_q}$
- 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

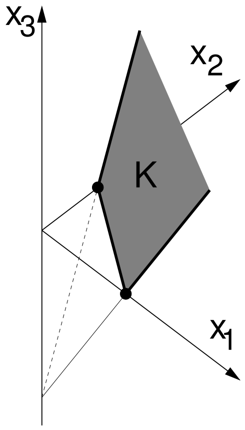
General LP problem is

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

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

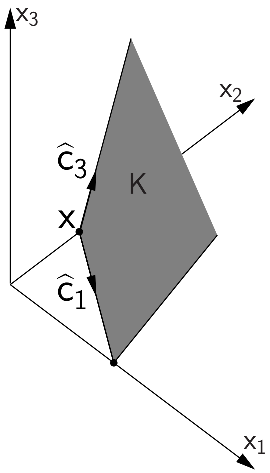
The feasible region

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



$n = 3;\ m = 1$

The simplex algorithm: Choosing where to move

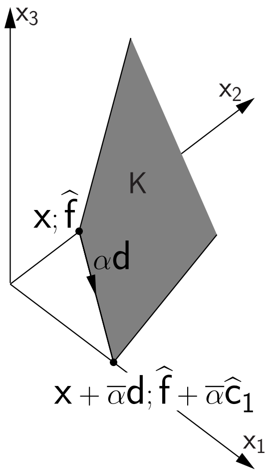


General LP problem is

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

- At a vertex $\boldsymbol{x}$, 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 $\widehat{c}_j$ of f with x_j is known
- If no positive $\widehat{c}_j$ then $\boldsymbol{x}$ is optimal so **stop**
- Increase variable x_q with greatest $\widehat{c}_j$

The simplex algorithm: Choosing how far to go



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

The simplex algorithm: Basic and nonbasic variables

At a vertex $\boldsymbol{x}$

- There are $n - m$ **nonbasic variables** $\boldsymbol{x}_N$ 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 $\boldsymbol{c}$ corresponding to $\mathcal{N}$ form the vector $\boldsymbol{c}_N$
- There are m **basic variables** $\boldsymbol{x}_B$ 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 $\boldsymbol{c}$ corresponding to $\mathcal{B}$ form the vector $\boldsymbol{c}_B$

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

$$\begin{array}{lcccccc} \text{maximize} & f & = & \boldsymbol{c}_B^T\boldsymbol{x}_B & + & \boldsymbol{c}_N^T\boldsymbol{x}_N & \\ \text{subject to} & & & B\boldsymbol{x}_B & + & N\boldsymbol{x}_N & = \boldsymbol{b} \\ & & & \multicolumn{3}{l}{\boldsymbol{x}_B \geq \boldsymbol{0},\ \boldsymbol{x}_N \geq \boldsymbol{0}} & \end{array}$$

The simplex algorithm: The reduced objective and optimality

- Using the partitioned equations

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

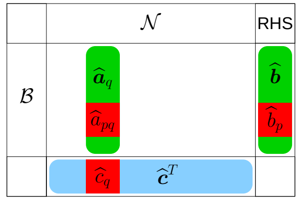
- Hence, substituting for $\mathbf{x}_B$, the objective function is

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

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

This is the key to solving LP problems

The simplex algorithm: Data requirements



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

Edge direction is thus

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

The simplex algorithm: Computational requirements

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

The revised simplex method

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

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

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

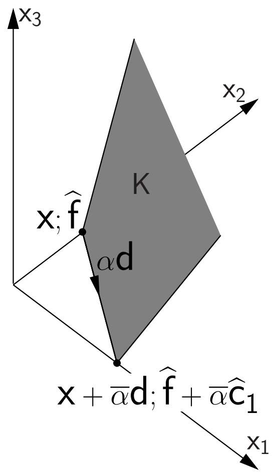
- Maintains $B^{-1}N$, $\widehat{\mathbf{b}}$ and $\widehat{\mathbf{c}}$ in a rectangular tableau
- Requires $O(mn)$ storage and $O(mn)$ computation per iteration
- Inefficient and prohibitively expensive for large problems

Revised simplex method

- Computes $\widehat{\mathbf{a}}_q = B^{-1}\mathbf{a}_q$ as required, forms $\widehat{\mathbf{c}}$ and updates $\widehat{\mathbf{b}} = B^{-1}\mathbf{b}$
- 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

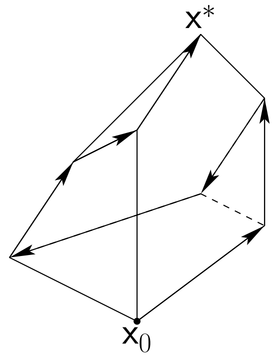


- Increase variable x_q with greatest $\hat{c}_j > 0$
- Edge direction $\boldsymbol{d}$ corresponding x_q is known
- Move along $\boldsymbol{x} + \alpha\boldsymbol{d}$
- Stop when first component of $\boldsymbol{x} + \alpha\boldsymbol{d}$ is zeroed
  x_q increases to $\overline{\alpha} > 0$
- Objective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

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

Proof of termination

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



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

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

Klee-Minty problems

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{j=1}^{n} 10^{j-1} x_j \\ \text{subject to} & x_i + 2\displaystyle\sum_{j=i+1}^{n} 10^{j-i} x_j \le 100^{n-i} \quad \text{for } i = 1, \ldots, n \\ & x_1 \ge 0, \ldots, x_n \ge 0 \end{array}$$

- 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 $\|\boldsymbol{d}_j\|$ 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 $\boldsymbol{c}_k^T\boldsymbol{x}_k$ s.t. $A_k\boldsymbol{x}_k = \boldsymbol{b}_k$ and $\boldsymbol{x}_k \ge \boldsymbol{0}$
- Person k consumes $x_{kj} = [\boldsymbol{x}_k]_j$ of food j so $\displaystyle\sum_{k=1}^{N} x_{kj} \le b_j$
- This **availability constraint** links the individual LPs

Dantzig-Wolfe structure

$$\begin{array}{llccccccc} \min & f & = & \boldsymbol{c}_1^T\boldsymbol{x}_1 & + & \ldots & + & \boldsymbol{c}_N^T\boldsymbol{x}_N & & \\ \text{s.t.} & & & A_{01}\boldsymbol{x}_1 & + & \ldots & + & A_{0N}\boldsymbol{x}_N & \le & \boldsymbol{b}_0 \\ & & & A_1\boldsymbol{x}_1 & & & & & = & \boldsymbol{b}_1 \\ & & & & & \ddots & & & & \vdots \\ & & & & & & & A_N\boldsymbol{x}_N & = & \boldsymbol{b}_N \\ & & & \boldsymbol{x}_1 \ge \boldsymbol{0} & & \ldots & & \boldsymbol{x}_N \ge \boldsymbol{0} & & \end{array}$$

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
- 69 years old and going strong!

Linear Programming 1: Summary

- 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