MAT3007 Optimization
Lecture 5 Geometry of LP
Simplex Method

Yuang Chen

School of Data Scienc
The Chinese University of Hong Kong, Shenzhen

June 12, 2025

Announcements

1. Next Monday (June 16)'s class is rescheduled to Tuesday (June 17) 1:30 – 3:20 pm via Zoom.

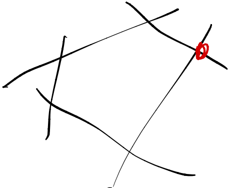
2. HW 1 due Sunday (June 15)
   HW 2 post today

   Codes into pdf file

Outline

Outline

Extreme Point



- Convex combination of m points: $\sum_{i=1}^{m} \lambda_i x_i$, $\sum_{i=1}^{m} \lambda_i = 1$, $\lambda_i \geq 0$, $\forall i$
- A point x in the polyhedron P is an **extreme point** of P if and only if x is not a convex combination of other two different points in P, i.e., there does not exist $y, z \in P$ $(x \neq y, x \neq z)$ and $\lambda \in [0, 1]$ so that $x = \lambda y + (1 - \lambda) z$.

Summary of Polyhedron Representations

$$P = \{x : Ax \le b\}$$

- Halfspace representation: A polyhedron is the intersection of a finite number of halfspaces. This representation applies to all polyhedra, bounded or unbounded

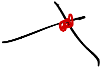
Polytope

- Extreme-point representation: A bounded polyhedron is the convex hull of all its extreme points. Convex hull can only generate a bounded set.

$$P = \{x : x = \sum_i \lambda_i x^i, \sum_i \lambda_i = 1, \lambda_i \ge 0\}$$

ex pt

Relation between extreme points and linearly independent active constraints



- An extreme point is a corner point where multiple constraints intersect, i.e. an extreme point is the intersection of multiple active constraints.
- Considering an LP with n variables, an extreme point is the intersection of n linearly independent active constraints. Since n linearly independent linear equations have a unique solution, an extreme point is the unique solution of the set of constraints active at this point.
- In other words, an extreme point x is the solution of a set of linear equations, which consist of linearly independent active constraints at x.

Outline

Basic solution

take "="

Consider a polyhedron defined by linear equalities and linear inequalities, and let x be an element of $\mathbb{R}^n$. The vector x is a **basic solution** if:

- There are n linearly independent, active constraints at x;
- All equality constraints are active at x.

Remark: BS may not be feasible.

Basic Feasible Solutions

Definition

If a basic solution x satisfies all constraints, then we call it a **basic feasible solution** (BFS).

To find a BFS

- First find a basic solution x
- Check if x satisfies all constraints

Theorem 1

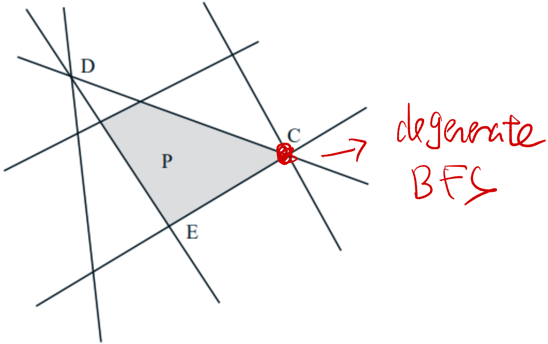
For the standard LP polyhedron $\{x : Ax = b, x \geq 0\}$, *the followings are equivalent:*

1. x *is an extreme point*
2. x *is a basic feasible solution*

Degeneracy

Degeneracy

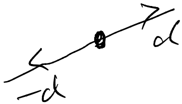
A basic feasible solution x^* is called degenerate if there are more than n active constraints at x^*.



Existence of BFS

Definitions: A polyhedron contains a line if $\exists x \in P$ and $d \in \mathbb{R}^n$, such that

$$x + \theta d \in P \quad \forall \theta$$



Theorem 2

$P = \{x \in \mathbb{R}^n | Ax \geq b\} \neq \Phi$. P has a BFS if and only if P does not contain a line.

$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \geq 0$

Corollary

- Polyhedron in standard form $P = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$ always has a BFS.
- Bounded polyhedron always has a BFS.

Optimality of BFS

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in P = \{x | Ax \geq b\} \end{aligned}$$

min. X
s.t. $X \leq 0$

Theorem 3

If P has at least one BFS, then LP is either unbounded or there exists a BFS which is optimal.

- In order to find an optimal solution, we only need to look among basic feasible solutions.
- LP is a convex optimization problem, which means local optimal is global optimal.

Outline

Standard Form LP

In the following, we consider LP in its standard form:

$$\begin{aligned} \min \quad & c^{\mathsf{T}} x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

m equality constraints

n nonnegative variable types

- $x \in \mathbb{R}^n$, i.e. there are n variables
- $A \in \mathbb{R}^{m \times n}$, i.e. there are m equality constraints
- We always assume all the m equality constraints are linearly independent (or equivalently A has full rank m), otherwise we can remove all redundant linearly dependent constraints or the problem is infeasible.
- Always assume $n > m$, i.e. more variables than constraints

To find a BFS, we first find a BS.

We need $\underline{n}$ LI constraints to find a BS.

$Ax=b$ gives me $\underline{m}$ LI constraints.

I need additional $\underline{n-m}$ constraints from $x \geq 0$. This means I need select $\underline{n-m}$ x's to be zero.

Basic Solution of Standard Form LP

- A basic solution is the unique solution to n linearly independent active constraints.
- For a standard form LP, we already have m linearly independent active constraints.
- Need $n - m$ additional linearly independent active constraints. Where to find them? From nonnegative constraints: $x_i \geq 0$. But which to choose to make active?

$n-m \quad x_j \quad = 0$

e.g.

$$\min.\ -x_1 - 2x_2$$

$$\text{s.t.}\ \begin{cases} x_1 + s_1 = 100 \\ 2x_2 + s_2 = 200 \\ x_1 + x_2 + s_3 = 150 \end{cases}$$

$$x_1, x_2, s_1, s_2, s_3 \geq 0$$

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

$$\begin{array}{c} \overbrace{\begin{array}{ccc} x_1 & x_2 & s_1 \end{array}}^{B} \ \ \overbrace{\begin{array}{cc} s_2 & s_3 \end{array}}^{N} \\ \left[\begin{array}{ccc:cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{bmatrix} \left.\begin{array}{c} x_1 \\ x_2 \\ s_1 \end{array}\right\} x_B \\ \left.\begin{array}{c} s_2 \\ s_3 \end{array}\right\} x_N \end{bmatrix} = b$$

$$\begin{array}{c} \overbrace{\begin{array}{ccc} x_2 & s_2 & s_3 \end{array}}^{B} \ \ \overbrace{\begin{array}{cc} x_1 & s_1 \end{array}}^{N} \\ \left[\begin{array}{ccc:cc} 0 & 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{array}\right] \end{array} \begin{bmatrix} \left.\begin{array}{c} x_2 \\ s_2 \\ s_3 \end{array}\right\} x_B \\ \left.\begin{array}{c} x_1 \\ s_1 \end{array}\right\} x_N \end{bmatrix} = b$$

$$A \qquad x = b$$

$$\|$$

$$\begin{array}{cc} & \begin{array}{cc} m & n-m \end{array} \\ m & \left[\begin{array}{c:c} B & N \end{array}\right] \end{array} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b$$

$A_B \leftrightarrow B \in \mathbb{R}^{m \times m}$: basis matrix

$A_N \leftrightarrow N \in \mathbb{R}^{m \times n-m}$: non-basis matrix

$x_B \in \mathbb{R}^{m}$: basic variables

$\rightarrow x_N \in \mathbb{R}^{n-m}$: non-basic variables

Basis: index of columns selected in B

$\rightarrow I_B$

m equations

($Ax=b$)

n equations

$$\begin{cases} Bx_B + Nx_N = b \\ x_N = 0 \end{cases}$$

n-m equations

$$\Downarrow$$

$$Bx_B = b$$

$$x_B = B^{-1} b$$

$$BS: \begin{cases} X_B = B^{-1}b & \text{(m elements)} \\ X_N = 0 & \text{(n-m elements)} \end{cases}$$

This BS must satisfy $Ax=b$

Check if $X_B = B^{-1}b \geq 0$, then this BS is a BFS.

Finding a Basic Solution in Standard Form LP

Procedures to find a basic solution:

1. Choose any m independent columns of A: $A_{B(1)}, \ldots, A_{B(m)}$ and form the basis matrix $B = [A_{B(1)}, \ldots, A_{B(m)}]$. Denote the rest of A as matrix N.
2. Let $x_i = 0$ for all $i \neq B(1), \ldots, B(m)$.
3. Solve the equation $Ax = b$ for the remaining $x_{B(1)}, \ldots, x_{B(m)}$.

- The basic solution is $x = [x_B, x_N]$, where the basic variables are $x_B = B^{-1}b$ and the nonbasic variables are $x_N = 0$.
- Since $A_{B(1)}, \ldots, A_{B(m)}$ are linearly independent, the last step must produce a unique solution.
- Basic solution of an LP only depends on its constraints, it has nothing to do with the objective function.

Find BS

$$\begin{aligned} \min \quad & c^\mathsf{T} x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

- $Ax = b$ gives m active constraints.
- Since $m < n$, need additional $n - m$ active constraints from $x \geq 0$.

$A = [B \quad | \quad N]$; $B \in \mathbb{R}^{m \times m}$ is **basis matrix** and $N \in \mathbb{R}^{m \times (n-m)}$

$x = (x_B, x_N)$; $x_B \in \mathbb{R}^m$ is **basic variables**

$x_N \in \mathbb{R}^{n-m}$ is **non-basic variables**

$Ax = b \to Bx_B + Nx_N = b$ variable → basis

$I_B = \{B(1), ..., B(m)\}$ is the **basic indices** for basic solution

The remaining indices I_N are the **non-basic indices**

Why does this method work?

- We can write the n active constraints as
  - $\begin{bmatrix} B & N \\ 0 & I \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$
- Since B is an invertible matrix, and I is identity matrix, the whole matrix is invertible, therefore, n active constraints are linearly independent
- Thus, there is only one solution, which is a basic solution
- The solution can be computed:
  - $Bx_B = b \Rightarrow \; x_B = B^{-1}b$
  - $x_N = 0$

Example

$$
\begin{array}{rrrl}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & x_1 & & \leq 100 \\
& & 2x_2 & \leq 200 \\
& x_1 & +x_2 & \leq 150 \\
& x_1, & x_2 & \geq 0
\end{array}
$$

The standard form:

$$
\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +s_1 & & & = 100 \\
& & 2x_2 & & +s_2 & & = 200 \\
& x_1 & +x_2 & & & +s_3 & = 150 \\
& x_1, & x_2, & s_1, & s_2, & s_3 & \geq 0
\end{array}
$$

Example Continued

We can write the feasible set by $\{x : Ax = b, x \geq 0\}$. where

$$A = \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \quad N \qquad b = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

Choose three independent columns of A, e.g., the first three, we get the corresponding basic solution is

$$x_N = \begin{bmatrix} s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ s_1 \end{bmatrix} = x_B = B^{-1}b = \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & s_1 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \end{array} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 50 \\ 100 \\ 50 \end{bmatrix}$$

That is $x_1 = 50, x_2 = 100, s_1 = 50$. Therefore $(50, 100, 50, 0, 0)$ is a basic feasible solution. One can find other basic feasible solutions by choosing other sets of columns. This is the first step of Simplex method.

Example Continued

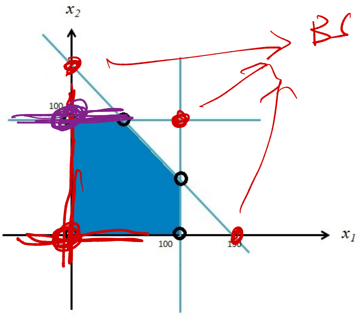
We can list the basic (feasible) solutions

Indices	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,2,5\}$	$\{1,3,4\}$
Solution	(50, 100, 50, 0, 0)	(100, 50, 0, 100, 0)	(100, 100, 0, 0, -50)	(150, 0, -50, 200, 0)
Status	BFS	BFS	Basic but not feasible	Basic but not feasible
Indices	$\{1,4,5\}$	$\{2,3,4\}$	$\{2,3,5\}$	$\{3,4,5\}$
Solution	(100, 0, 0, 200, 50)	(0, 150, 100, -100, 0)	(0, 100, 100, 0, 50)	(0, 0, 100, 200, 150)
Status	BFS	Basic but not feasible	BFS	BFS

The other two choices $\{1,3,5\}$ and $\{2,4,5\}$ lead to dependent basic columns (therefore no basic solutions can be obtained)

Verify

They indeed correspond to all the corners of the feasible sets.



BS not feasible

Quiz

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} \begin{array}{l} \to m \\ \to n-m \end{array}$$

How many non-zeros could one have in a basic solution (assuming there are m constraints)?

- No more than m
- Could be anything between 0 to m, but typically it is m

How many basic solutions can one have for a linear program with m constraints and n variables?

- At most $C(n, m) = \frac{n!}{m!(n-m)!} = \binom{n}{m}$ (Combination number)
- Therefore for a finite number of linear constraints, there can only be a finite number of basic solutions

Search Among BFS

Now we know that we only need to search among basic feasible solutions for the optimal solution.

How to search among the basic feasible solutions?

- One may suggest to list all the basic feasible solutions and compare their objective values. However, there are too many of them.
- For a linear optimization with m constraints and n variables, how many basic feasible solutions it may have?
- $C(n, m)$.. If $n = 1000$, $m = 100$, then the value is 10^{143}..

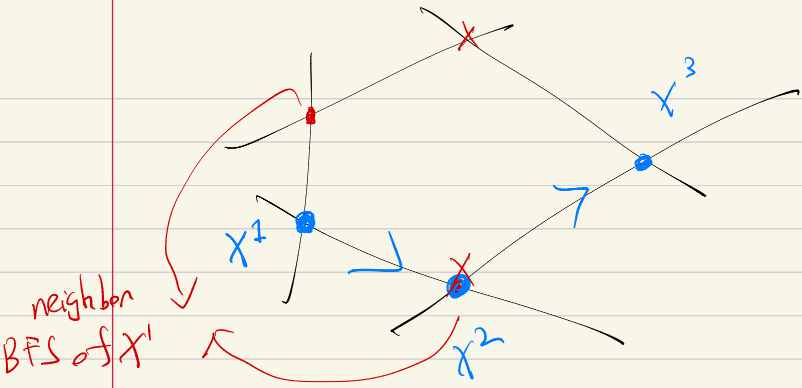
Simplex Method

Therefore we need a smarter way to find the optimal solution.

- Simplex method

The simplex method proceeds from one BFS (a corner point of the feasible region) to a neighboring one, in such a way as to continuously improve the value of the objective function until reaching optimality.

- We need to define what it means by *adjacent* or *neighboring* solution
- We need to design an efficient way to find (and move to) the neighboring BFS (e.g., we should try to avoid taking matrix inversions every time)
- We need to design a valid stopping criterion

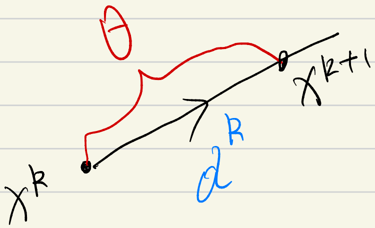


k: iteration number

$$x^{k+1} \leftarrow x^k + \theta \, d^k$$

stepsize

direction
$d^k \in R^n$



θ is a positive scalars
$\theta \in R^+$

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} + \theta \begin{bmatrix} d_B \\ d_N \end{bmatrix}$$

Outline

Feasible Direction - Maintain Feasibility

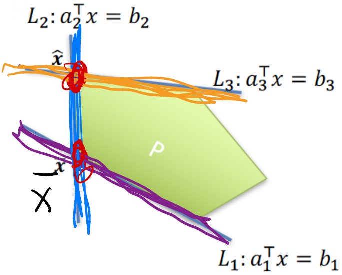
$x + \theta d$ is feasible

Starting from a basic feasible solution x, the simplex method considers a feasible direction d to move away from the BFS x to $\hat{x} := x + \theta d$. The new point $x + \theta d$ needs to be (a) a feasible point and (b) an adjacent BFS. For (a), we need

$$\begin{aligned} & A(x + \theta d) = b \\ \Rightarrow \quad & Ax + \theta A d = b \\ \Rightarrow \quad & \underline{Ad = 0} \end{aligned}$$

$$Ad = \begin{bmatrix} B & N \end{bmatrix} \begin{bmatrix} d_B \\ d_N \end{bmatrix} = Bd_B + Nd_N = 0$$

- Definition: Two basic feasible solutions x and $\hat{x}$ of a polyhedron P are called adjacent if they share the same $n-1$ linearly independent active constraints.



Feasible region P:

$$a_1^\top x \le b_1$$
$$a_2^\top x \le b_2$$
$$a_3^\top x \le b_3$$
$$a_4^\top x \le b_4$$
$$a_5^\top x \le b_5$$

At $\bar{\boldsymbol{x}}$, L_1 and L_2 are L.I.A.C.
At $\hat{\boldsymbol{x}}$, L_2 and L_3 are L.I.A.C.
$\bar{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}$ are adjacent BFS's
and they share 1 L.I.A.C. (L_2)

Adjacent BFS

- Standard form LP

$$\begin{aligned} \min \quad & c^{\mathsf{T}} x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

$x_N = 0$

- Two adjacent BFS differ the basis matrix B in exactly one column
- Two adjacent BFS differ by exactly one basic (or non-basic) index.
- In a standard form LP, two BFS x and $\hat{x}$ are adjacent if they have the same $n - m - 1$ nonbasic variables, and differ in one nonbasic variable.
- Because $n - m - 1$ nonbasic variables of x need to remain nonbasic, i.e. at zero value, d_N must have n-m-1 components at zero value; and because one nonbasic variable of x needs to become basic, i.e. to increase from zero value to some positive value, then the corresponding component of d_N has to be a positive number.

Adjacent BFS

- Let I_B be the basic variable indices and I_N be the nonbasic variable indices.
- We want to select a nonbasic variable x_j, $j \in I_N$ (remember initially $x_j = 0$) to enter the basis: increase x_j to a positive number while keeping other nonbasic variables at zero.
- $d_N = e_j^\top = [0, ..., 0, 1, 0, ...0]^\top$ for some $d_j = 1$.

$$d_N = \overbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}^{e_j} \quad 1 \leftrightarrow x_j$$

$$
\begin{aligned}
& B d_B + N \overbrace{d_N}^{e_j} = 0 \\
\Rightarrow \quad & B d_B + A_j = 0 \\
\Rightarrow \quad & d_B = -B^{-1} A_j
\end{aligned}
$$

Put together, we have $d = \begin{bmatrix} -B^{-1} A_j \\ e_j \end{bmatrix}$. We refer to this direction as the **j-th basic direction**.

e.g. $x = \begin{array}{c c} \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} & \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \end{array} \quad x_B = \begin{bmatrix} x_1 \\ x_2 \\ s_2 \end{bmatrix} \quad x_N = \begin{bmatrix} s_1 \\ s_3 \end{bmatrix}$

Select s_3 enter basis (increase)

$j = s_3$

$$d_N = \begin{bmatrix} d_{s_1} \\ d_{s_3} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = e_{s_3}$$

remark

$\times e_2$

$$d = \begin{bmatrix} d_B \\ d_N \end{bmatrix} = \begin{bmatrix} d_{x_1} \\ d_{x_2} \\ d_{s_2} \\ d_{s_1} \\ d_{s_3} \end{bmatrix} = \begin{bmatrix} -B^{-1}A_{s_3} \\ 0 \\ 1 \end{bmatrix}$$

Adjacent BFS Example

- If $\mathbf{x} = (x_1, \ldots, x_5)$ has nonbasic variables $x_3 = x_4 = x_5 = 0$, then its adjacent BFS must share two of these three nonbasic variables, i.e., $x_3 = x_4 = x_2 = 0$ may be nonbasic variables in an adjacent BFS.
- This means d_N contains 2 zeros and only 1 one component:

$$d_N = \begin{bmatrix} d_3 \\ d_4 \\ d_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} dx_3 \\ dx_4 \\ dx_5 \end{bmatrix}$$

- Then $\mathbf{x} + \theta \mathbf{d}$ will make x_5 positive, i.e., increasing from zero.

Choosing a Direction and Pivoting

n-m non-basic variables.

- At the current BFS $\bar{x}$, each nonbasic variable x_j provides a direction:

$$d_N = e_j \quad \text{and} \quad d_B = -B^{-1}A_j,$$

pointing to an adjacent BFS.

$\hat{x} = x + \theta d$

- Which direction should the algorithm pick?
- The algorithm should pick a direction to reduce objective cost.
- How does the objective value change along a direction?
  - $\underbrace{c^\top(x+\theta d)}_{} - \underbrace{c^\top x}_{}$ is the change of objective value
  - $c^\top d$ is the change of objective value for a unit stepsize
  - $\theta c^\top d$ is the total change of objective value after moving θd
- The algorithm should pick a **d** such that $c^\top d < 0$
- This is called **pivoting**: make x_j enter the basis

Current: x

Next: $x+\theta d$

$$\text{Objective change} = \overbrace{c^T(x+\theta d)}^{\text{smaller}} - \overbrace{c^T x}^{\text{larger}}$$

$$= \underbrace{\theta}_{\geq 0} c^T d$$

We want $c^T d < 0$

$$\bar{c}_j < 0$$

Cost Change

$$c = \begin{bmatrix} c_B \\ c_N \end{bmatrix}$$

$$c^T d$$

$$\begin{aligned} c^{\mathsf{T}}(x + \theta d) - c^{\mathsf{T}} x &= \theta \begin{bmatrix} c_B^{\mathsf{T}} & c_N^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} d_B \\ d_N \end{bmatrix} \\ &= \theta(c_B^{\mathsf{T}} d_B + c_N^{\mathsf{T}} \cancel{d_N}) \quad e_j \\ &= \theta(\underbrace{-c_B^{\mathsf{T}} B^{-1} A_j + c_j}_{\text{reduced cost}}) \quad c^T d \end{aligned}$$

$$= \bar{c}_j$$

Reduced Cost

For each $j \in I_N$, we define the **reduced cost** $\bar{c}_j$ of the variable x_j to be $\bar{c}_j = c_j - c_B^{\mathsf{T}} B^{-1} A_j$

Theorem: Optimality Conditions

4

Consider a basic feasible solution x associated with a basis matrix B, and let $\bar{c}$ be the corresponding vector of reduced costs.

- If $\bar{c}_j \geq 0$ for all $j \in I_N$, then x is optimal. → stopping criterion
- If x is optimal and nondegenerate, then $\bar{c}_j \geq 0$ for all $j \in I_N$

Thus, we want to pick $j \in I_N$ such that the reduced cost $\bar{c}_j < 0$.

- This theorem gives a stopping criterion to the simplex algorithm: We stop when all the reduced costs are non-negative.
- It also means that if one could not find a neighbor solution that is better, then one must have already achieved optimal solution.

$$\begin{array}{lrrrrrl} \text{minimize} & -x_1 & -2x_2 & & & & \\ \text{subject to} & x_1 & & +s_1 & & & = 100 \\ & & 2x_2 & & +s_2 & & = 200 \\ & x_1 & +x_2 & & & +s_3 & = 150 \\ & x_1, & x_2, & s_1, & s_2, & s_3 & \geq 0 \end{array}$$

If we are at basis $\{1, 4, 5\}$ with BFS $(100, 0, 0, 200, 50)$. Then the reduced costs are:

x_1, s_2, s_3

$X_N = \begin{bmatrix} x_2 \\ s_1 \end{bmatrix}$

$$\bar{C}_{x_1} = \bar{c}_2 = -2 - \underbrace{[-1, 0, 0]}_{C_B} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1}}_{B} \underbrace{\begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}}_{A_{x_2}} = -2$$

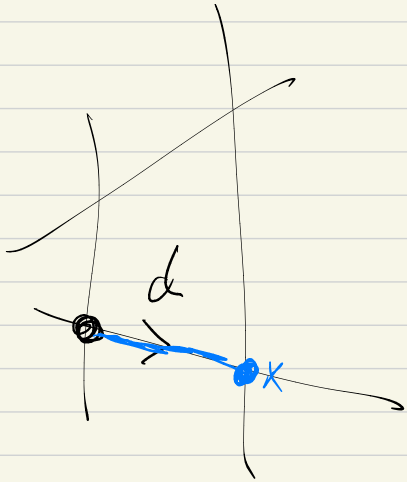
$\bar{C}_{s_1}$ "

Similarly, $\bar{c}_3 = 1$. Therefore including x_2 in the basis in the next step will reduce the objective value.

$$\min. \quad -x_1 - 2x_2$$

$$c = \begin{array}{c} \begin{bmatrix} -1 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{array} \begin{array}{l} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{array} \qquad x_B = \begin{bmatrix} x_1 \\ s_2 \\ s_3 \end{bmatrix}$$

$$c_B = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$



Change of Basis

Now we know that at certain x, if the reduced costs $\bar{c}_j \geq 0$, then x is optimal. What if some $\bar{c}_j < 0$?

- Then it means that by bringing jth variable (non-basic) into the basis, we can decrease the objective value. Thus we want to go in that direction

Assume d is the jth basic direction with $\bar{c}_j < 0$. We know that going in this direction can reduce the objective. But how much can we go?

- We need to make sure that $x + \theta d \geq 0$ to maintain feasibility.
- We also want to go as far as possible
- Therefore, we choose

$$\theta^* = \max\{\theta \geq 0 | x + \theta d \geq 0\}$$

Stepsize

$x_i + \theta d_i \geq 0$

$$\theta^* = \max\{\theta \geq 0 | \underline{x + \theta d} \geq 0\}$$

$\theta \leq -\frac{x_i}{d_i}$

- If $d \geq 0$ (specifically $d_B = -B^{-1}A_j \geq 0$), then $\theta^* = \infty$. In this case, one can go unlimitedly far without making the solution infeasible, while keeping the objective decreasing. Therefore, the original LP is unbounded
- If $d_i < 0$ for some $i \in I_B$, then we can solve:

$$\theta^* = \min_{\{i \in I_B | d_i < 0\}} \left(-\frac{x_i}{d_i} \right)$$

The optimal basic variable index $i \in I_B$ that achieves the min corresponds to $x_i + \theta^* d_i = 0$, i.e. the basic variable x_i exists the basis, becoming a nonbasic variable.

New solution: j enters and i exits basis

$k \in I_N$

$$y_k = \begin{cases} 0 & k \in I_N \backslash j \\ \theta^* & k = j \end{cases}$$

$k \in I_B$

$$y_k = \begin{cases} x_k + \theta^* d_k & k \in I_B \backslash i \\ 0 & k = i \end{cases}$$

An Iteration of the Simplex Method

We start from a BFS $\boldsymbol{x}$ (with corresponding basis B)

1. We first compute the reduced costs $\bar{\boldsymbol{c}}$ for all nonbasic variables

$$\bar{c}_j = c_j - \boldsymbol{c}_B^T B^{-1} A_j$$

   - If no reduced costs is negative, then $\boldsymbol{x}$ is already optimal
   - Otherwise choose some j such that $\bar{c}_j < 0$

2. Compute the jth basic direction $\boldsymbol{d} = \begin{bmatrix} -B^{-1}A_j \\ e_j \end{bmatrix}$
   - If $\boldsymbol{d} \geq 0$, then the problem is unbounded.
   - Otherwise, compute $\theta^* = \min_{i \in I_B, d_i < 0} \{-\frac{x_i}{d_i}\}$

3. Let $\boldsymbol{y} = \boldsymbol{x} + \theta^* \boldsymbol{d}$. Then $\boldsymbol{y}$ is the new BFS with index j replacing i in the basis, where i is the index attaining the minimum in θ^*. Objective value is changed by $\theta^* \boldsymbol{c}^T \boldsymbol{d} = \theta^* \bar{c}_j$.

4. Simplex method repeats these procedures until one stopping criteria is met.

Outline

Degeneracy

In most of the cases, the objective value will strictly decrease after one simplex method iteration. However, it is possible that the objective stays the same.

Since the change of the objective value (if one chooses to have x_j enter the basis) is $\theta^* \bar{c}_j$ and we know that $\bar{c}_j < 0$. This can only happen if $\theta^* = 0$.

Recall that

$$\theta^* = \min_{\{i \in I_B | d_i < 0\}} \left(-\frac{x_i}{d_i} \right)$$

If for i's such that $d_i < 0$, there exists $x_i = 0$, then $\theta^* = 0$. This may happen when there are 0s in the BFS $\boldsymbol{x}$.

Degeneracy

Definition (Degeneracy)

We call a basic feasible solution $\boldsymbol{x}$ degenerate if some of the basic variables are 0.

- Degeneracy could happen. As an algorithm, we need to consider what consequences it may have

An example:

$$\begin{array}{cccccccc} x_1 & +2x_2 & +x_3 & & & = & 8 \\ x_1 & -x_2 & & +x_4 & & = & 4 \\ -x_1 & +x_2 & & & +x_5 & = & 4 \\ x_1 & , x_2 & , x_3 & , x_4 & , x_5 & \geq & 0 \end{array}$$

If we choose the basic indices to be $\{1, 2, 4\}$, then the corresponding basic solution is $(0, 4, 8)$. It is therefore degenerate.

- The is equivalent to that the number of non-zeros at a basic solution is strictly less than m

Degeneracy

Assume degeneracy happens at some point:

- Given a BFS $\boldsymbol{x}$ with negative reduced cost $\bar{c}_j < 0$ and $\theta^* = 0$. And i is the index that achieves $\min_{\{i \in I_B, d_i < 0\}} (-x_i/d_i)$. Thus, $x_i = 0$.

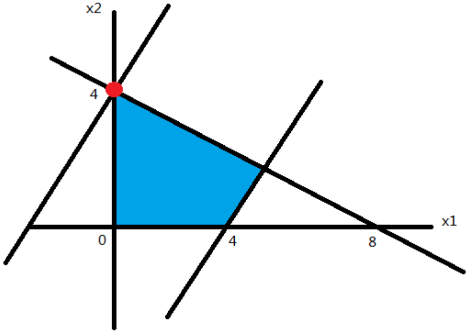
We can still view that we changed the basis from i (i leaving the basis) to j (j entering the basis) and proceed to the next iteration.

- Although the solution (and the objective value) does not change, the basis changed. Therefore, the reduced costs vector will change in the next iteration — issue seems resolved

However, we need to guarantee that there won't be any cycle, i.e., we will not visit the same BFS more than once

- This can only happen together with degeneracy, since otherwise the objective value will strictly decrease

Illustration



- More than 2 lines intersect at one point

Example of Cycling

If not dealt properly, cycle can happen. Consider the following LP:

$$A = \begin{pmatrix} -2 & -9 & 1 & 9 & 1 & 0 \\ 1/3 & 1 & -1/3 & -2 & 0 & 1 \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$\boldsymbol{c} = (-2, -3, 1, 12, 0, 0)$$

If we set $B = \{5, 6\}$ initially, then the sequence shown below leads to a cycle (objective value doesn't change, and there is always an index with negative reduced cost):

Step #	1	2	3	4	5	6
Exiting	x_6	x_5	x_2	x_1	x_4	x_3
Entering	x_2	x_1	x_4	x_3	x_6	x_5
Basis Index	$(5,2)$	$(1,2)$	$(1,4)$	$(3,4)$	$(3,6)$	$(5,6)$

We will show that cycle can be avoided by designing how to choose incoming/outgoing basis when there are multiple choices.

Pivoting Rules: Choose the Entering Basis

In the description of the algorithm, we say that at each feasible solution, we can choose *any* j with negative reduced cost to enter the basis in the next iteration. Sometimes, there are more than one j with $\bar{c}_j < 0$. In this case, we need to make some rules to choose the entering basis.

Here are several possible rules:

1. *Smallest subscript rule*: choose the smallest index j such that $\bar{c}_j < 0$
2. *Most negative rule*: choose the smallest $\bar{c}_j$
3. *Most decrement rule*: choose j with the smallest $\theta^* \bar{c}_j$

Pivoting Rules: Choose the Exiting Basis

Recall that

$$\theta^* = \min_{\{i \in I_B | d_i < 0\}} \left(-\frac{x_i}{d_i} \right)$$

And we choose the index that attains this minimum to leave the basis. It is possible that there are two or more indices that attain the minimum (tie). Then we also need a rule to decide the outgoing basis.

- The most commonly used rule is the *smallest index rule*

When this tie happens, the next BFS will be degenerate

Bland's Rule

Theorem (Bland's Rule)

If we use both the smallest index rule for choosing the entering basis and the exiting basis, then no cycle will occur in the simplex algorithm.

Using the Bland's rule when applying the simplex method, we can guarantee to stop within a finite number of iterations at an optimal solution.