DDA5002 Optimization
Lecture 4 Simplex Method

Yuang Chen

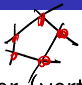
School of Data Science
The Chinese University of Hong Kong, Shenzhen

September 24, 2025

Announcement

- Homework 2 will be posted tomorrow and is due Sunday, October 12.
- We have two lectures before National Break: September 24 and 28. We do have a makeup class in the evening on Sunday.

Recap



- The optimal solution of LP will be at a corner (vertex) of the feasible region.
- Definition of polyhedron: $P = \{x \in \mathbb{R}^n : Ax \leq b\}$: intersection of many halfspaces.
- 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 $x = \lambda y + (1 - \lambda)z$.
- A set of constraints are **linearly independent** if the normal directions (vectors) of constraints are linearly independent.
- When two or more linearly independent constraints are active/binding at a certain point (take the equal "=" sign), then we call them **linearly independent constraints active/binding at this point**.

" = "

Recap

- $x \in \mathbb{R}^n$ is a **basic solution** of a polyhedron if
  - There are n linearly independent constraints active at x;
  - All equality constraints are active at x.
- If a basic solution x satisfies all constraints, then we call it a **basic feasible solution** (BFS).
- Corner points (vertices), basic feasible solutions, and extreme points are equivalent.
- If a polyhedron is nonempty and has at least one extreme point (BFS), then the LP is either unbounded or there exists an extreme point (BFS) which is optimal.

Outline

Outline

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

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

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$\begin{bmatrix} - a_1^\mathsf{T} - \\ - a_2^\mathsf{T} - \\ \vdots \\ - a_m^\mathsf{T} - \end{bmatrix} x = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

- $x \in \mathbb{R}^n$, i.e. there are n variables
- $A \in \mathbb{R}^{m \times n}$, i.e. there are m equality constraints
- Assumption I: all the m equality constraints are linearly independent (or equivalently A has linearly independent rows or A has full row rank m). If not, we can remove all redundant linearly dependent constraints.
- Assumption II: $n > m$, i.e. more variables than equality constraints.

We need n equations.

$Ax=b$ has m equations

I still need $n-m$ equations from $x \geq 0$.

$\Downarrow$

Select $n-m$ x variables to set to zero.

$x_N \in R^{n-m}$ nonbasic variables.

$x \in R^n$, $x_N \in R^{n-m}$,

Put the remaining m x variables into

$x_B \in R^m$ basic variables

$$\begin{cases} \text{basic variables} & x_B \geq 0 \quad \text{Positive} \\ \text{nonbasic variables} & x_N = 0 \quad \text{zero} \end{cases}$$

Different! basic solution $\rightarrow$ $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

basic variable $\rightarrow$ x_i

Basic Solution of Standard Form LP

- A basic solution is the unique solution to n linearly independent constraints taking equal signs (n equations).
- For a standard form LP, we already have m linearly independent equations ($Ax = b$).
- Need additional $n - m$ linearly independent constraints. Where to find them? From nonnegative constraints: $x \geq 0$. But which x_i to choose?
- Selecting $n - m$ variables to set to zero is equivalent to selecting the remaining m variables to be basic (potentially nonzero).

positive

Basic Variables and Non-basic Variables

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

- $Ax = b$ gives m constraints.
- Since $m < n$, need additional $n - m$ constraints from $x \geq 0$.
- Select $n - m$ variables from x and put in a vector x_N.
- $x_N \in \mathbb{R}^{n-m}$ are **non-basic variables**.
- Let $N = \{N(1), \ldots, N(n-m)\}$ denote the indices in x_N called **non-basic indices**.
- Put the remaining m variables in a vector x_B.
- $x_B \in \mathbb{R}^m$ are **basic variables**.
- Let $B = \{B(1), \ldots, B(m)\}$ denote the indices in x_B called **basic indices** or **basis**.

In total, x has n variables.

Select $n-m$ x variables as X_N

$\Updownarrow$

Select m x variables as X_B

Step I

Key Findings

- For a linear system of equations $Ax = b$, a row of A corresponds to a specific constraint.
- For a linear system of equations $Ax = b$, i-th column of A corresponds to specific variable x_i.
- We can switch the columns of A as long as they correspond to the order sequence of the variables in x.

$$
\begin{array}{ccc} x_1 & x_2 & x_3 \\ \end{array}
$$
$$
\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}
$$

row $\leftrightarrow$ constraint

column $\leftrightarrow$ variable

$$
\begin{array}{ccc} x_3 & x_1 & x_2 \\ \end{array}
$$
$$
\begin{bmatrix} 3 & 1 & 2 \\ 6 & 4 & 5 \\ 9 & 7 & 8 \end{bmatrix} \begin{bmatrix} x_3 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}
$$

$$Ax = b$$

Rewrite $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$

e.g. $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$

$x_B = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix}$ $x_N = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} \rightarrow x = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ x_2 \\ x_4 \end{bmatrix}$

$$A = \begin{bmatrix} | & | & | & | & | \\ A_1 & A_2 & A_3 & A_4 & A_5 \\ | & | & | & | & | \end{bmatrix}$$

$$A_B = \begin{bmatrix} | & | & | \\ A_1 & A_3 & A_5 \\ | & | & | \end{bmatrix} \qquad A_N = \begin{bmatrix} | & | \\ A_2 & A_4 \\ | & | \end{bmatrix}$$

$A_B \in \mathbb{R}^{m \times m}$ basis matrix corresponding to x_B

$A_N \in \mathbb{R}^{m \times n-m}$ nonbasis matrix corresponding to x_N

$$A = [\ A_B \quad A_N\ ]$$

e.g.

$$\begin{array}{cc} \overbrace{}^{A_B} & \overbrace{}^{A_N} \end{array}$$

$$\begin{bmatrix} | & | & | & | & | \\ A_1 & A_3 & A_5 & A_2 & A_4 \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ x_2 \\ x_4 \end{bmatrix} \begin{matrix} \} x_B \\ \\ \} x_N \end{matrix} = b$$

$$\Updownarrow$$

$$Ax = b$$

Basis Matrix

After rearranging the variables, we have

- $x = [x_B, x_N]$ where $x_B \in \mathbb{R}^m$ are **basic variables** and $x_N \in \mathbb{R}^{n-m}$ are **non-basic variables**.
- Select the columns of A corresponding to basic variables and put them in a matrix A_B, called **basis matrix**.
- $$A_B = \begin{bmatrix} | & | & & | \\ A_{B(1)} & A_{B(2)} & \cdots & A_{B(m)} \\ | & | & & | \end{bmatrix}$$
- The remaining columns of A corresponding to non-basic vairables are **the non-basic columns**. Put them in a matrix A_N, called **non-basis matirx**.
- After rearranging columns of A, we have $A = [A_B \quad A_N]$.

We are finding a BS, we need n equations

$$\left.\begin{cases} \underline{Ax = b} & - \ m \text{ equations} \\ x_N = 0 & - \ n-m \text{ equations} \end{cases}\right\} n \text{ equations}$$

$$A = [\, A_B \;\vdots\; A_N \,] \qquad x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$$

$$[A_B \quad A_N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b$$

$$\Rightarrow A_B x_B + A_N x_N = b \qquad Ax=b$$

$$\begin{cases} A_B x_B + A_N x_N = b \\ x_N = 0 \end{cases} \Rightarrow A_B x_B = b$$

$$\Downarrow$$

$$\begin{cases} x_B = A_B^{-1} b \\ x_N = 0 \end{cases} \rightarrow \text{BS}$$

Logic to Find a Basic Solution

- We can write the n equations as

$$\begin{bmatrix} A_B & A_N \\ 0 & I \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \rightarrow \begin{cases} A_B X_B + A_N X_N = b \\ I X_N = 0 \end{cases}$$

- Since A_B is an invertible matrix, and I is the identity matrix, the whole matrix is invertible; therefore, the n equations are linearly independent.
- Thus, there is only one solution, which is a basic solution.
- The solution can be computed:

$$A_B x_B = b \Rightarrow x_B = A_B^{-1} b$$
$$x_N = 0$$
$$\rightarrow BS$$

Finding a Basic Solution in Standard Form LP

Procedures to find a basic solution:

select m x variables $\to x_B$

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

   $x_B = A_B^{-1} b$

- The basic solution is $x = [x_B, x_N]$, where the basic variables are $x_B = A_B^{-1}b$ and the nonbasic variables are $x_N = 0$.
- Since $A_{B(1)}, ..., A_{B(m)}$ 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.

check!

BS: $\begin{cases} x_B = A_B^{-1} b \overset{?}{\geq} 0 \\ x_N = 0 \end{cases}$ feasible?

$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$ ✓

Check Whether BS is a BFS?

- The constraint $Ax = b$ must be satisfied.
- Since $x_N = 0$, the nonbasic variables must be nonnegative.
- **Check feasibility:** If $x_B = A_B^{-1}b \geq 0$, then the basis solution is a basic feasible solution (BFS).

Example

$m = 3$
$n = 5$

slack variables

$$
\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = 100 \\
 & & 2x_2 & & +x_4 & & = 200 \\
 & x_1 & +x_2 & & & +x_5 & = 150 \\
 & x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0
\end{array}
$$

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

Example

Choose the first three columns of A as the basis matrix:

$$A_B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \quad A_N = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$x_B = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = A_B^{-1} b = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 50 \\ 100 \\ 50 \end{bmatrix} \geq 0$$

$$\mathbf{x}_N = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Therefore $(50, 100, 50, 0, 0)$ is a basic solution. Since all basic variables are nonnegative ($x_B \geq 0$), thus it is a basic feasible solution.

Example

Select columns 2, 3, and 4 of A as the basis:

$$A_B = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$x_B = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = A_B^{-1} b = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 150 \\ 100 \\ -100 \end{bmatrix} < 0$$

$$\mathbf{x}_N = \begin{bmatrix} x_1 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since $x_4 = -50 < 0$, the basis solution is not feasible. This is a basic solution but not a basic feasible solution (BFS).

Example

Select columns 3, 4, and 5 of A as the basis:

$$A_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}$$

$$x_B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = A_B^{-1} b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} \geq 0$$

$$\mathbf{x}_N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since $x_B \geq 0$, the basic solution $(0, 0, 100, 200, 150)$ is a basic feasible solution (BFS).

Example Continued

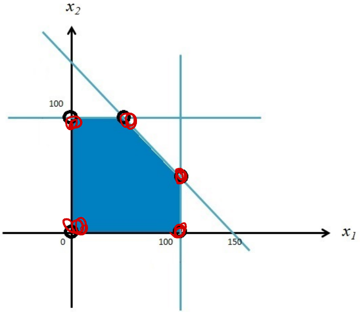
We can list the basic (feasible) solutions

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)

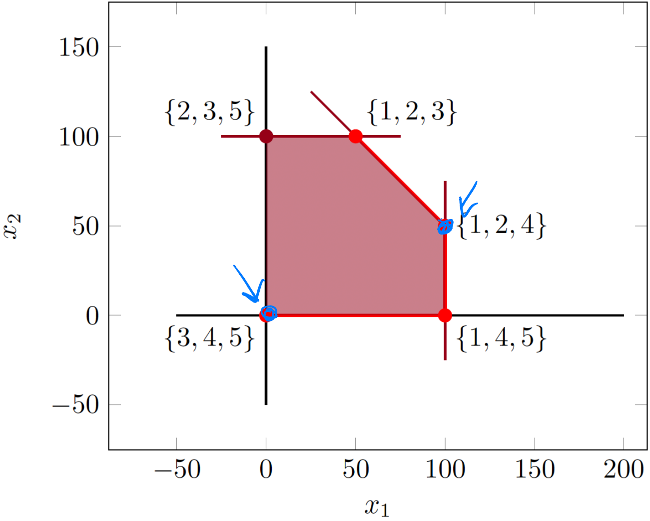
Verify

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



Verify

Each BFS correspond to a selection of columns (basis matrix).



Quiz

$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

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

BS: $\begin{cases} x_B = A_B^{-1} b & (m) \\ x_N = 0 & (n-m) \end{cases}$

- No more than m ↗ x_B
- 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 $C(n, m) = \frac{n!}{m!(n-m)!}$ (Combination number) > $\binom{n}{m}$
- Therefore for a finite number of linear constraints, there can only be a finite number of basic solutions

Exercise: How many BFS can a standard form LP have?

Theorems for Standard Form LP

Theorem (LP fundamental theorem)

Given a linear optimization problem in standard form where A has full row rank m

1. *If there is a feasible solution, there is a basic feasible solution;*
2. *If there is an optimal solution, there is an optimal solution that is a basic feasible solution.*

Corollary

If an LP with m constraints (in the standard form) has an optimal solution, then there must be an optimal solution such that there is no more than m positive entries.

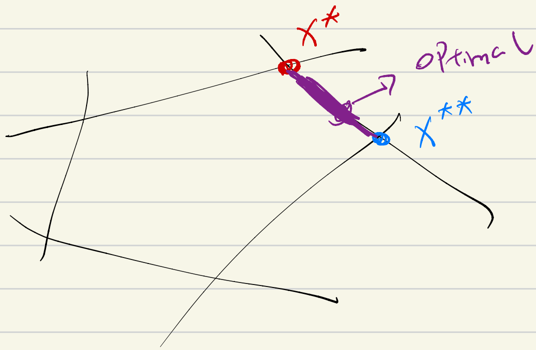
BFS $\to$ BS ! $\begin{cases} x_B = A_B^{-1} b \quad (m) \\ x_N = 0 \end{cases}$

Remark:

For an LP with at least one optimal soln,

(1) there must be an optimal soln at a BFS. ✓

(2) any optimal soln is a BFS ✗



Proof Sketch

We first prove part (i). Assume there is a feasible solution x. Let B denote the index set for which x is positive, and let k be the size of B. Then we show that if x is not a basic solution, we can always find a solution that has an index set with size less than k. By repeatedly doing this we must be able to find a basic feasible solution.

To show that, if x is not a basic solution, then it must be that A_i, $i \in B$ are linearly dependent. Suppose $\sum_{i \in B} \alpha_i A_i = 0$ where at least one α_i is positive (we can always achieve that). And we define a vector α by

$$\alpha = \begin{cases} \alpha_i & i \in B \\ 0 & \text{otherwise} \end{cases}.$$

Then we consider $x - \epsilon\alpha$. There must exist an ϵ such that it is still feasible and that there is one more 0 entry in $x - \epsilon\alpha$. Thus part (i) is proved.

Proof Sketch Continued

Now we consider part (ii). We continue to use the arguments for part (i). Let x be an optimal solution with k positive entries. If x is not a basic solution, then we show that we can always find an optimal solution with fewer than k positive entries (or reach a contradiction). By repeatedly doing this, we must be able to find an optimal basic feasible solution.

We continue to define the same α. We claim that it must be that $c^T\alpha = 0$, otherwise, one can construct a better solution than x by considering $x + \epsilon\alpha$ and $x - \epsilon\alpha$ for some small α, which contradicts with the optimality of x.

If $c^T\alpha = 0$, then $x - \epsilon\alpha$ are still optimal solutions. And by previous arguments, we can find one optimal solution with fewer positive entries.

Search Among BFS

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

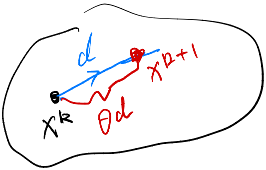
How to search among the basic feasible solutions?

- 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}.

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

Outline

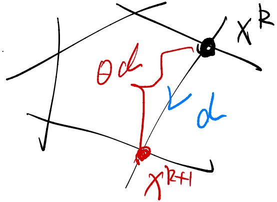
Basic Structure of an Optimization Algorithm



At each iteration k,

1. Start from a feasible solution $\mathbf{x}^k$.
2. Find a "good" direction $\mathbf{d}$ that (a) points inside the feasible region and (b) decreases the objective value.
3. Find a "good" step length θ along $\mathbf{d}$ to move to next iteration point: $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \theta\mathbf{d}$.
4. If no good direction or step length can be found, terminate. Otherwise $k \leftarrow k + 1$ and go back to step 1.

Simplex Method Framework



At each iteration k,

1. Start from a *basic feasible solution* $\mathbf{x}^k$.
2. Find a direction $\mathbf{d}$ that (a) points to an *adjacent BFS* and (b) decreases the objective value.
3. Find a step length θ so that the next iteration point, $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \theta\mathbf{d}$, is a better adjacent BFS.
4. If no such direction or step length can be found, terminate. Otherwise $k \leftarrow k + 1$ and go back to step 1.

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

Outline

Simplex Method Starting Point

First, we assume that we have somehow found a BFS whose basis is

$$B = \{B(1), \dots, B(m)\}.$$

Define:

$$A_B = \begin{bmatrix} | & | & & | \\ A_{B(1)} & A_{B(2)} & \cdots & A_{B(m)} \\ | & | & & | \end{bmatrix}$$

and let A_N be the matrix consisting of the non-basic columns of A.

In the sequel, N will denote the non-basic index set.

Rearranging the variables, we can write

$$A = [A_B, \, A_N], \qquad x = \begin{bmatrix} x_B \\ x_N \end{bmatrix},$$

where x_B are the basic variables and x_N are the non-basic variables.

By definition, we have: $$x_B = A_B^{-1} b \geq 0, \quad x_N = 0.$$ $\rightarrow$ BFS

Current BFS and Update Step

Let the current basic feasible solution be:

$$\mathbf{x} = \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}$$

Next iterate:

$$\hat{\mathbf{x}} = \mathbf{x} + \theta \mathbf{d} = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix} + \theta \begin{bmatrix} d_2 \\ d_3 \\ d_5 \\ d_1 \\ d_4 \end{bmatrix} \begin{array}{l} \\ d_B \\ \\ \\ d_N \end{array}$$

Direction Vector Partition

Direction vector $\mathbf{d}$ is partitioned as:

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \\ d_1 \\ d_4 \end{bmatrix}, \quad \mathbf{d}_B = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \end{bmatrix}, \quad \mathbf{d}_N = \begin{bmatrix} d_1 \\ d_4 \end{bmatrix}$$

Here, $\mathbf{d}_B$ corresponds to the basic variables (x_2, x_3, x_5), and $\mathbf{d}_N$ corresponds to the nonbasic variables (x_1, x_4).

Feasible Direction - Maintain Feasibility

$$y = x + \theta d$$

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

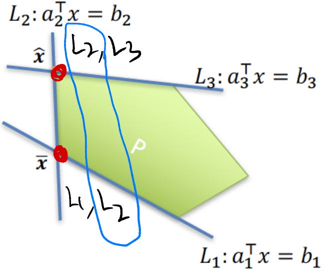
$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

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

$$Ad = [A_B \;\; A_N] \begin{bmatrix} d_B \\ d_N \end{bmatrix} = A_B d_B + A_N d_N = 0$$

Outline

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

different

1 equation

$n-1$ equations

1 equation

x

$\hat{x}$

Standard form LP

$$\begin{cases} Ax=b \\ x\geq 0 \end{cases}$$

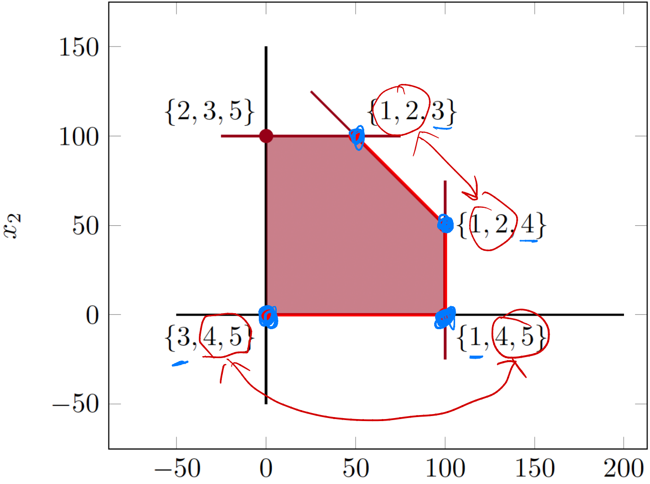
To find a BFS / BS

$$\begin{cases} \underline{Ax=b} \\ x_N=0 \end{cases}$$ must satisfy

1 different

Adjacent BFS in Standard Form LP

Two basic solutions are neighboring / adjacent if they differ by exactly one basic (or non-basic) index.



Adjacent BFS in Standard Form LP

- Standard form LP

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

- Two adjacent BFS differ the basis matrix A_B in exactly one column.
- Two adjacent BFS differ by exactly one basic (or non-basic) variable.
- 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.

Two adjacent BFS differ in

- one column in basis matrix A_B
- one basic variable
- one index in basis B
- one column in non-basis matrix A_N
- one non-basic variable
- one index in N

In total, there are $n-m$ nonbasic variables.

Two adjacent BFS differ in 1 nonbasic variable.

share $n-m-1$ nonbasic variables.

$\Downarrow$

Set as zero

directions for them are zero

Move from nonbasic to basic variable

Move from 0 to positive

direction for this variable is 1

Non-basic Direction

move from nonbasic variable to basic variable

- To find a neighbor, we want to select a nonbasic variable x_j, $j \in N$ (remember initially $x_j = 0$) to enter the basis: increase x_j to a positive number while keeping other nonbasic variables at zero.
- We consider moving x (the current BFS) to a neighboring one y by $y = x + \theta d$.
- For nonbasic variables, we need $d_j = 1$ for some $j \in N$ and $d_{j'} = 0$ for all other non-basic indices $j' \neq j, j' \in N$.
- $d_N = e_j^T = [0, ..., 0, 1, 0, ...0]^T$ for some $d_j = 1, j \in N$.

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = e_j$$

x_j

Basic Direction

To maintain feasible, we need

e_j

$$
\begin{aligned}
& A_B d_B + A_N d_N = 0 \\
\Rightarrow \quad & A_B d_B + A_j = 0 \\
\Rightarrow \quad & d_B = -A_B^{-1} A_j
\end{aligned}
$$

Put together, we have
$d = \begin{bmatrix} d_B \\ d_N \end{bmatrix} = \begin{bmatrix} -A_B^{-1} A_j \\ e_j \end{bmatrix} = [-A_B^{-1} A_j; 0; \ldots; 1; 0; \ldots; 0]$. The direction d is uniquely determined once j is chosen. We refer to this direction as the **j-th basic direction**.

$A_N d_N$

$j \longleftrightarrow N(3)$

$= A_N e_j$

$$= \begin{array}{cccc} x_{N(1)} & x_{N(2)} & x_{N(3)} & x_{N(4)} \\ \end{array}$$

$$\begin{bmatrix} | & | & | & | \\ A_{N(1)} & A_{N(2)} & A_{N(3)} & A_{N(4)} \\ | & | & | & | \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$= A_{N(3)} = A_j$

$x_N = \begin{bmatrix} x_2 \\ x_4 \\ x_6 \end{bmatrix}$ $\quad A_N = [A_2 \;\; A_4 \;\; A_6]$

$j = 4$

$A_N e_4$

$= [A_2 \;\; A_4 \;\; A_6] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $\neq e_4$

wrong!

$= [A_2 \;\; A_4 \;\; A_6] \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $= e_4$

x_4

$= A_4$

Example - Basic Directions

$$
\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = 100 \\
& & 2x_2 & & +x_4 & & = 200 \\
& x_1 & +x_2 & & & +x_5 & = 150 \\
& x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0
\end{array}
$$

Current basis: $\{x_2, x_3, x_5\}$, corresponding to columns 2, 3, and 5 of A.

$$
A_B = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}
$$

The current basic feasible solution is

$$
\mathbf{x}_B = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 50 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
$$

Example - Basic Directions

- The nonbasic variables $x_1 = x_4 = 0$, then its adjacent BFS must share one of these two nonbasic variables, i.e., $x_1 = x_2 = 0$ may be nonbasic variables in an adjacent BFS. Let's select nonbasic variable x_4 to enter the basis.
- This means d_N contains 1 zero and 1 one component:

$$d_N = \begin{bmatrix} d_1 \\ d_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \neq e_2 = e_4$$

x_4

- Then $\mathbf{x} + \theta \mathbf{d}$ will make x_4 positive, i.e., increasing from zero.
- Compute the direction for basic variables:

x_4

$$d_B = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \end{bmatrix} = -A_B^{-1} A_4 = -\begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Outline

Objective Function Partition

The linear objective is: $= c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4 + c_5 x_5$

$$\mathbf{c}^T\mathbf{x} = \begin{bmatrix} \underbrace{c_2 \quad c_3 \quad c_5}_{c_B} & \underbrace{c_1 \quad c_4}_{c_N} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix} \begin{array}{l} x_B \\ \\ x_N \end{array}$$

We partition $\mathbf{c}$ and $\mathbf{x}$ into basic and nonbasic parts:

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix} = \begin{bmatrix} c_2 \\ c_3 \\ c_5 \\ c_1 \\ c_4 \end{bmatrix}, \quad \mathbf{c}_B = \begin{bmatrix} c_2 \\ c_3 \\ c_5 \end{bmatrix}, \quad \mathbf{c}_N = \begin{bmatrix} c_1 \\ c_4 \end{bmatrix}$$

$= c_2 x_2 + c_3 x_3 + c_5 x_5 + c_1 x_1 + c_4 x_4$

So the objective becomes: $[c_B \;\; c_N]^T \begin{bmatrix} x_B \\ x_N \end{bmatrix} =$

$$\mathbf{c}^T\mathbf{x} = \cancel{\begin{bmatrix} \mathbf{c}_B & \mathbf{c}_N \end{bmatrix}} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N$$

Choosing a Direction and Pivoting

- At the current BFS, each nonbasic variable x_j provides a direction:

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

  pointing to an adjacent BFS.

$\underset{\text{next}}{y} = \underset{\text{current}}{x} + \theta d$

- 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? $\theta c^\top d < 0$
  - $c^\top(x + \theta d) - c^\top x = \theta c^\top d$ 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 $c^\top d < 0$
- Selecting a $j \in N$ such that $c^\top d < 0$ is called **pivoting**: make x_j enter the basis

Cost Change

$$c^\top y - c^\top x = c^\top(x+\theta d) - c^\top x = \theta c^\top d$$

$y = x + \theta d$

$$= \theta \begin{bmatrix} c_B^\top & c_N^\top \end{bmatrix} \begin{bmatrix} d_B \\ d_N \end{bmatrix}$$

$$= \theta(c_B^\top d_B + c_N^\top d_N)$$

$$= \theta(\underbrace{-c_B^\top A_B^{-1} A_j + c_j}_{\text{reduced cost}}) < 0$$

Reduced Cost

For each $j \in N$, we define the **reduced cost** $\bar{c}_j$ of the variable x_j to be $\bar{c}_j = c_j - c_B^\top A_B^{-1} A_j$.

- We want to select a nonbasic variable x_j such that the reduced cost $\bar{c}_j < 0$, which means the objective value will decrease.

$$c^T d$$

$$= c_B^T d_B + c_N^T d_N$$

$$= c_B^T(-A_B^{-1}A_j) + c_N^T d_N$$

$$= -c_B^T A_B^{-1} A_j + c_N^T e_j$$

$$= -c_B^T A_B^{-1} A_j + c_j$$

$$= c_j - c_B^T A_B^{-1} A_j < 0$$

I want

reduced cost for nonbasic variable x_j

$$\begin{cases} d_B = -A_B^{-1}A_j \\ d_N = e_j \end{cases}$$

e.g. $x_N = \begin{bmatrix} x_2 \\ x_4 \\ x_6 \end{bmatrix}$

$j = 4$

$$[c_2 \ c_4 \ c_6]\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= c_4$$

$$= c_j$$

Quiz

What is the reduced cost of a basic variable?

$$\bar{c}_j = c_j - c_B^T \underline{A_B^{-1} A_j}, \quad j \in B$$

$$= c_j - c_B^T e_j$$

$$= c_j - c_j = 0$$

$j \in B$

$A_B^{-1} A_j$

$= e_j$

$A_B^{-1} A_B = I$

$$\begin{bmatrix} A_B^{-1} \end{bmatrix}\begin{bmatrix} A_j \end{bmatrix} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

Optimality Conditions in Simplex

Optimality Conditions in Simplex

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

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

- Thus, we want to pick $j \in N$ 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.

Proof of First Part

Let x be a BFS with $\bar{c} \geq 0$. Let z with $Az = b$ and $z \geq 0$ be an arbitrary feasible point.
Define $u = z - x$. We have $Au = 0$. That is

$$0 = [A_B \ \ A_N] \begin{bmatrix} u_B \\ u_N \end{bmatrix} = A_B u_B + \sum_{j \in N} A_j u_j.$$

This gives

$$u_B = -\sum_{j \in N} A_B^{-1} A_j u_j.$$

Let us now compute

$$c^T u = [c_B^T \ c_N^T] \begin{bmatrix} u_B \\ u_N \end{bmatrix} = c_B^T u_B + \sum_{j \in N} c_j u_j = \sum_{j \in N} (c_j - c_B^T A_B^{-1} A_j) u_j = \sum_{j \in N} \bar{c}_j u_j.$$

Notice that $\bar{c}_j \geq 0$, $\forall j$ and $u_j = z_j - x_j = z_j \geq 0$, $\forall j \in N$. We have

$$c^T u = c^T z - c^T x \geq 0$$

for any arbitrary feasible z.

Example: Reduced Costs

$$c_B = \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \rightarrow x_2 \\ \rightarrow x_3 \\ \rightarrow x_5 \end{matrix}$$

$$\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = 100 \\
 & & 2x_2 & & +x_4 & & = 200 \\
 & x_1 & +x_2 & & & +x_5 & = 150 \\
 & x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0
\end{array}$$

$$x_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}$$

If we are at basis $\{2, 3, 5\}$ then the reduced costs are:

$$c_1 - c_B^T A_B^{-1} A_1$$

$$\bar{c}_1 = -1 - \begin{bmatrix} -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = -1 \quad < 0$$

$$c_4 - c_B^T A_B^{-1} A_4$$

$$\bar{c}_4 = 0 - \begin{bmatrix} -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 1 \quad > 0$$

Therefore only including x_1 in the basis in the next iteration will reduce the objective value.

Example: Reduced Costs – Continued

$$
\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = 100 \\
& & 2x_2 & & +x_4 & & = 200 \\
& x_1 & +x_2 & & & +x_5 & = 150 \\
& x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0
\end{array}
$$

If we are at basis $B = \{1, 2, 3\}$ with BFS $(50, 100, 50, 0, 0)$. Then the reduced costs are

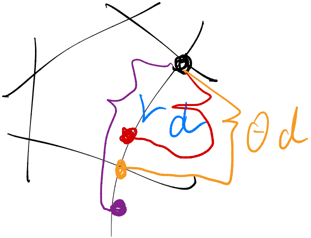
$x_N = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}$

$$
\bar{c}_4 = 0 - [-1, -2, 0] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0.5 > 0
$$

$$
\bar{c}_5 = 0 - [-1, -2, 0] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 1 > 0
$$

Therefore, the reduced costs are all nonnegative and the current BFS is optimal.

Outline



Change of Basis and Choose Stepsize θ

At the current BFS x, if there exists a $j \in N$ such that $\bar{c}_j < 0$:

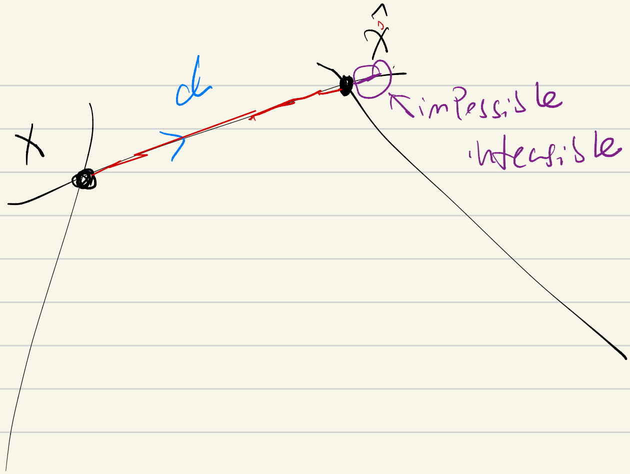
- It means that by bringing the non-basic variable x_j into the basis, we can decrease the objective value. Thus, we want to go in that direction (i.e., compute d as the jth basic direction).

$y = x + \theta d$ $Ay = b$ ✓ $y \geq 0$?

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 until hitting the adjacent BFS.
- Therefore, we choose

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



$$\max.\ \theta$$
$$\text{s.t. } x + \theta d \geq 0 \rightarrow x_k + \theta d_k \geq 0$$
$$\theta \geq 0$$

For nonbasic variables $\overset{=0}{x_j}$, $j \in N$

$d_N = e_j$ $\quad \underline{x_j + \theta d_j \geq 0} \Rightarrow x_j + \theta \begin{cases} 0 = x_j = 0 \\ 1 = x_j + \theta = \theta \geq 0 \end{cases}$

For basic variables x_i, $i \in B$

$d_B = -A_B^{-1} A_j$ $\quad x_i + \theta d_i \geq 0 \Rightarrow x_i - \theta (A_B^{-1} A_j)_i \rightarrow$ negative

$$\overset{\geq 0}{x_i} + \overset{\geq 0}{\theta}\, \overset{\geq 0}{d_i}$$

$d_i \geq 0$ impossible to be negative

$d_i < 0$ Possible to be negative.

I only need to consider $i \in B$, $d_i < 0$ case

e.g.

$$\max.\ \theta$$

$$\text{s.t. } 3 - 4\theta \geq 0 \rightarrow \theta \leq \frac{3}{4}$$

$$2 - \theta \geq 0 \rightarrow \theta \leq 2$$

$$6 - 2\theta \geq 0 \rightarrow \theta \leq 3$$

$$\theta^* = \frac{3}{4}$$

$$\max.\ \theta$$

where $d_i < 0$

$$\text{s.t. } x_i + \theta d_i \geq 0, \quad \forall i \in B : d_i < 0$$

$$\theta d_i \geq -x_i$$

$$\boxed{\theta \leq -\frac{x_i}{d_i} \quad \forall i \in B : d_i < 0}$$

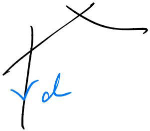
$$\Downarrow$$

$$\theta \leq \min_{i \in B, d_i < 0} \left\{ -\frac{x_i}{d_i} \right\}$$

$$\theta^* = \min_{i \in B, d_i < 0} \left\{ -\frac{x_i}{d_i} \right\}$$

Min-ratio Test

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



$d_N = e_j \geq 0$

- If $d \geq 0$ (specifically $d_B = -A_B^{-1} A_j \geq 0$), 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 $i \in B$, then we can solve $x_i + \theta d_i = 0$ for some $\{i \in B : d_i < 0\}$. Then, taking minimum gives:

$$\theta^* = \min_{\{i \in B : d_i < 0\}} \frac{-x_i}{d_i}. \rightarrow \text{min-ratio test}$$

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

$$\theta^* = \min_{i \in B, d_i < 0} -\frac{x_i}{d_i}$$

i^* achieves the min of $-\frac{x_i}{d_i}$ θ^*

$$x_{i^*} + \theta^* d_{i^*}$$

$$= x_{i^*} - \frac{x_{i^*}}{d_{i^*}} \cdot d_{i^*} = x_{i^*} - x_{i^*} = 0$$

x_{i^*} basic variable ≥ 0 $\rightarrow$ nonbasic variable $= 0$

x_{i^*} exit the basis

Outline

New solution: j enters and i exits basis

Simplex iteration:

$$y = x + \theta^* d$$ keep

let x_j enter the basis

$k \in N$

nonbasic

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

become basic

$k \in B$

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

let x_i exit the basis

- Basis for current iteration BFS x:

exit

$$B = \{B(1), \ldots i, B(m)\}$$

- Basis for next iteration BFS y:

enter

$$B' = \{B(1), \ldots, B(\ell - 1), j, B(\ell + 1), \ldots, B(m)\}$$

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

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

   - If $\bar{\boldsymbol{c}} \geq 0$, then $\boldsymbol{x}$ is already optimal
   - Otherwise choose some j such that $\bar{c}_j < 0$

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

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. Repeat 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 $\bar{c}_j < 0$. The objective stays the same 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 $\boldsymbol{x}$.

Definition (Degeneracy)

We call a basic feasible solution x **degenerate** if some of the basic variables are 0. Otherwise, it is **non-degenerate**.

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

An example:

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

If we choose the basic indices to be $\{1, 2, 4\}$, then the corresponding basic solution is $x_B = (x_1, x_2, x_4) = (0, 4, 8)$. It is therefore degenerate.

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

$X_B = A_B^{-1} b > 0 \rightarrow$ non-degenerate

Impact of Degeneracy on Simplex

- Suppose at the start of the current iteration, the BFS is degenerate:
  - $x = [0,2,3,0,0]^\top$, $x_B = \overset{x_1, x_2, x_3}{[0,2,3]^\top}$, $x_N = \overset{x_4, x_5}{[0,0]^\top}$
- Suppose the simplex method found a descent direction:
  - $d = [-1, -1, 2, 1, 0]^\top$
  - That is, x_4 enters the basis
- By min-ratio test:
  - $\min\left\{\frac{0}{1}, \frac{2}{1}\right\} = \underset{\theta^*}{0}$ so x_1 exists the basis
- The new BFS:
  - $x = [0,2,3,0,0]^\top$, $x_B = \overset{x_2, x_3, x_4}{[2,3,0]^\top}$, $x_N = \overset{x_1, x_5}{[0,0]^\top}$

The new BFS is the same point as the starting BFS! Simplex method Stayed at the same point. This can only happen to degenerate BFS.

The simplex method is updating in this iteration: basis changes!

Degeneracy

Assume degeneracy happens at some point:

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

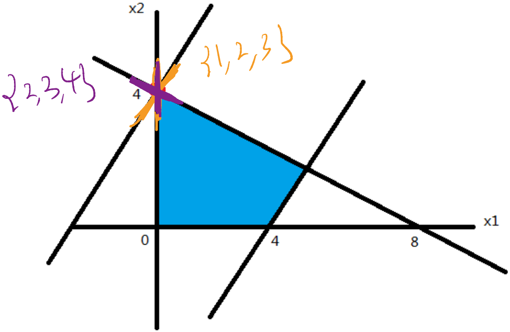
- Although the solution (and the objective value) does not change, the basis changes. Therefore, the reduced costs vector will change in the next iteration — issue seems resolved

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

$\{1,2,3\} \to \{2,3,4\} \to \{1,2,3\} \to \{2,3,4\}$



- 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

Select $j \in N$ s.t. $\bar{c}_j < 0$

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

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

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. (Why?)

Bland's Rule

choosing from x_2, x_6, x_8

$\downarrow$

x_2

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, or find the LP is unbounded.

Outline

Finding an Initial BFS

In our previous discussion, we assumed that we start with a certain BFS

- This can be done easily if the standard form is derived by adding slacks to each constraint and the right hand side is all nonnegative. (Why?)

However, in general, it is not necessarily easy to get an initial BFS from the standard form. For example,

$$
\begin{array}{lrrrrcr}
\text{minimize} & x_1 & +x_2 & +x_3 & & & \\
\text{subject to} & x_1 & +2x_2 & +3x_3 & & = & 3 \\
& & -4x_2 & -9x_3 & & = & -5 \\
& & & 3x_3 & +x_4 & = & 1 \\
& x_1 & ,x_2 & ,x_3 & ,x_4 & \geq & 0
\end{array}
$$

Finding an Initial BFS

- One could test different basis B, to see if $A_B^{-1}b \geq 0$.
- However, this may take a long time.
- In fact, in terms of computational complexity (which we will define later), finding one BFS is as hard as finding the optimal solution!

We will discuss an initialization method next — two-phase method.

Two-Phase Simplex Method

In the two-phase simplex method, we first solve an auxiliary problem ($\boldsymbol{e}$ means an all-one vector).

$$\begin{aligned} \min_{x,y} \quad & e^T y \\ \text{s.t.} \quad & Ax + y = b \\ & x, y \geq 0 \end{aligned}$$

Without loss of generality, we assume $b \geq 0$ (otherwise, we pre-multiply that row by -1).

There is a trivial BFS to the auxiliary problem: $(x = 0, y = b \geq 0)$ so one can apply the Simplex method to solve it.

Theorem

The original problem is feasible if and only if the optimal value of the auxiliary problem is 0.

Proof

For the " $\Longrightarrow$ " side, for any x feasible to the original problem, i.e.,

$$Ax = b,\ x \geq 0.$$

Then, it is easy to observe that $(x, y = 0)$ is feasible to the auxiliary problem. This immediately gives that $(x, y = 0)$ is also optimal to the auxiliary problem, as the function value of the auxiliary problem is always non-negative. For the " $\Longleftarrow$ " side, suppose that $(\hat{x}, \hat{y} = 0)$ is optimal to

the auxiliary problem. Then, we have

$$A\hat{x} + \hat{y} = A\hat{x} = b,$$

$$\hat{x} \geq 0.$$

This means that $\hat{x}$ is feasible to the original problem.

Two-Phase Simplex Method

By above Theorem, we can solve the auxiliary problem by the Simplex method, and

1. If the optimal value is not 0, then we can claim that the original problem is infeasible;
2. If the optimal value is 0 with solution $(x^* = \hat{x}, y^* = 0)$. Then we know that $\hat{x}$ must be a BFS for the auxiliary problem. Then it must be a BFS for the original problem as well. And we can start from there to initialize the simplex method.

Degenerate Case

If there is at least one basic variable in $\hat{y}$ (degenerate case), $\hat{x}$ will not contain the full basis B. What can we do in this case?

Let us assume one basic index is in $\hat{y}$ (say $B(m)$) w.l.o.g.:

- We still have $A\hat{x} = b$, $\hat{x} \geq 0$ (feasibility of $\hat{x}$).
- $\hat{x}$ has $m-1$ basic variables corresponding to $x_{B(1)}, \ldots, x_{B(m-1)}$. In addition, other parts in $\hat{x}$ must be zeros since they are non-basic.
- Then one can pick any other column from A (such that it forms a full-rank matrix with $A_{B(1)}, \ldots, A_{B(m-1)}$) in the non-basic part in $\hat{x}$ to make it a BFS for the original problem.

Procedure of the Two-Phase Method

Phase I:

1. Construct the auxiliary problem such that $b \geq 0$.
2. Solve the auxiliary problem using the Simplex method.
   - If we reach an optimal solution with optimal value greater than 0, then the original problem is infeasible.
3. If the optimal value is 0 with optimal solution $(x^* = \hat{x}, y^* = 0)$, then we enter phase II.

Phase II: Solve the original problem starting from the BFS $\hat{x}$.

- If $(\hat{x}, \hat{y} = 0)$ for the auxiliary problem has one or more basic indices appearing in $\hat{y}$, then we need to supplement some indices from the nonbasic part in $\hat{x}$ to make $\hat{x}$ a BFS for the original problem.

The Big-M method

There is another method that can be used to solve LP without a starting BFS. Consider the following auxiliary problem:

$$\begin{aligned} \min_{x,y} \quad & c^T x + M \textstyle\sum_{i=1}^m y_i \\ \text{s.t.} \quad & Ax + y = b \\ & x, y \geq 0 \end{aligned}$$

This problem has an initial BFS $x = 0, y = b \geq 0$ (assuming $b \geq 0$). Now we can use simplex to solve it. In the simplex procedure, pretend that M is a very large value (larger than any specified number).

- If the original problem is feasible, then optimal y must be 0.