MAT3007 Optimization
Lecture 6 Simplex Method

Yuang Chen

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

June 17, 2025

Agenda

- Tuesday (June 17): Simplex Method
- Wednesday (June 18): Simplex Tableau
- Thursday (June 19): LP Duality Formulation
- Monday (June 23): LP Duality Theorems and Applications
- Wednesday (June 25): Sensitivity Analysis, Midterm Review
- Thursday June 26: Midterm

Midterm

Final

HW 2 Posted and is due next Tuesday.

Outline

Outline

Standard Form LP

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

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \qquad \text{(m equality constants)} \\ & x \geq 0 \qquad \text{(n nonnegative constants)} \end{aligned}$$

- $x \in \mathbb{R}^n$, 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 $\underline{n > m}$, i.e. more variables than constraints

Example

$$\begin{array}{llll} \text{maximize} & x_1 & +2x_2 & \\ \text{subject to} & x_1 & & \le 100 \\ & & 2x_2 & \le 200 \\ & x_1 & +x_2 & \le 150 \\ & x_1, & x_2 & \ge 0 \end{array}$$

The standard form:

$$\begin{array}{lllllll} \text{minimize} & -x_1 & -2x_2 & s_1 & & & \\ \text{subject to} & 1\,x_1 & 0 & +x_3 & s_2 & & = 100 \\ & 0\,x_1 & 2x_2 & & +x_4 & s_5 & = 200 \\ & 1\,x_1 & +x_2 & & & +x_5 & = 150 \\ & x_1, & x_2, & x_3, & x_4, & x_5 & \ge 0 \end{array}$$

$m=3$

$n=5$

min. $c^T x$
s.t. $Ax=b$
$x \geq 0$

Example in Matrix Representation

$$A = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}, \quad \mathbf{b} = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

$Ax = b$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example in Matrix Representation

$$A' = \begin{array}{c} \begin{array}{ccccc} x_2 & x_4 & x_1 & x_5 & x_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}, \quad \mathbf{b} = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_5 \\ x_3 \end{bmatrix} \quad \mathbf{c}' = \begin{bmatrix} -2 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

$$c'^T x' = -x_1 - 2x_2$$

$$A'x' = b$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_5 \\ x_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

$$\begin{cases} x_1 + x_3 = 100 \\ 2x_2 + x_4 = 200 \\ x_2 + x_1 + x_5 = 150 \end{cases}$$

Example in Matrix Representation

$$A'' = \begin{array}{c} \begin{array}{ccccc} x_3 & x_4 & x_5 & x_1 & x_2 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}, \quad \mathbf{b} = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

$$A''x'' = b$$

$$\mathbf{x}'' = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{c}'' = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -2 \end{bmatrix}$$

$$c''^T x'' = -x_1 - 2x_2$$

Outline

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 from $Ax = b$.
- Need $n - m$ additional linearly independent active constraints. Where to find them? From nonnegative constraints: $x \geq 0$.
- We select $n - m$ variables to set to zero, which is equivalent to selecting the remaining m variables to be basic (potentially nonzero).
- These m basic variables correspond to a set of linear independent columns of A that form a basis (matrix).

Basic solution: n linearly independent active ('=')
constraints active at this point

min. $c^T x$

s.t. $Ax = b$ (m LI equations)

$x \geq 0$

I want to solve $x_1, \cdots, x_n$, n unknowns,
I need n equations. Currently, I already
have m equations. So, I still need $n-m$
equations, from $x \geq 0$.

Thus, I need $n-m$ x's set to zero.

$$\begin{cases} Ax = b \rightarrow m \text{ equations} \\ x_i = 0, \text{ for } n-m \; x_i\text{'s} \end{cases}$$

n equations

Find BS

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

- Select m linearly independent columns of A to form the basis matrix B; the remaining $n - m$ columns form the matrix N.
- The basis matrix B corresponds to the basic variables x_B, and the matrix N corresponds to the nonbasic variables x_N.
- $A = [B \quad | \quad N]$, where $B \in \mathbb{R}^{m\times m}$ is basis matrix and $N \in \mathbb{R}^{m\times(n-m)}$.
- $x = (x_B, x_N)$ where $x_B \in \mathbb{R}^m$ is **basic variables** and $x_N \in \mathbb{R}^{n-m}$ is **non-basic variables**.

non basic variables

Select $n-m$ $x_i = 0$

Potentially

Select the remaining m $x_i \neq 0$

basic variable

m x_i

$$A = \left[\begin{array}{c:c} B & N \end{array} \right]$$

$$X = \begin{bmatrix} X_B \\ X_N \end{bmatrix}$$

$X_B \in R^m$: basic variables

$X_N \in R^{n-m}$: non basic variables

$$A = [\, B \mid N \,]$$

$B \in R^{m \times m}$ basis matrix

$N \in R^{m \times n-m}$ non-basis matrix

same m equations

$$Ax = b$$

$$[\, B \mid N \,] \begin{bmatrix} X_B \\ X_N \end{bmatrix} = b$$

$$BX_B + NX_N = b$$

m equations

Find BS

$$Ax=b$$

n equations
$$\begin{cases} BX_B + NX_N = b & \to \text{m equations} \\ X_N = 0 & \to \text{n-m equations} \end{cases}$$

$$BX_B = b \Rightarrow X_B = B^{-1}b$$

One BS: $\begin{cases} X_B = B^{-1}b \\ X_N = 0 \end{cases}$

You select:

- m basic variables (X_B)
- B matrix
- n-m nonbasic variables (X_N)
- N matrix

basis

they are equivalent

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:
  - $Bx_B = b \Rightarrow \; x_B = B^{-1}b$
  - $x_N = 0$

Finding a Basic Solution in Standard Form LP

Procedures to find a basic solution:

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

- 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 $A_{B(1)}, ..., A_{B(m)}$ 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 Whether BS is a BFS?

BS, $\begin{cases} x_B = B^{-1}b \\ x_N = 0 \end{cases}$ may not be feasible

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

Example

$$
\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = 100 \\
 & & 2x_2 & & +x_4 & & = 200 \\
 & x_1 & +x_2 & & & +x_5 & = 150 \\
 & x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0
\end{array}
\quad \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} m=3 \quad \rightarrow n=5
$$

$x_B \in \mathbb{R}^3$, $x_N \in \mathbb{R}^2$

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

Example

select x_1, x_2, x_3 as basic variables

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

$$B = \begin{array}{ccc} x_1 & x_2 & x_3 \\ \end{array}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

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

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

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 x_2, x_3, x_4 as basic variables

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

$$B = \begin{array}{c} \begin{array}{ccc} x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array}, \quad N = \begin{array}{c} \begin{array}{cc} x_1 & x_5 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \end{array}$$

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

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

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:

$$B = \begin{array}{c} \begin{array}{ccc} x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}, \quad N = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} \end{array}$$

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

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

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

$$A = \left[\begin{matrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{matrix} \; \boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}\right] \quad b = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

Select x_3, x_4, x_5 as basic variables

$$x_B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad B = \begin{array}{c} \begin{matrix} x_3 & x_4 & x_5 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \quad N = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} \end{array} \quad x_N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Select x_5, x_3, x_4 as basic variables

$$B' = \begin{array}{c} \begin{matrix} x_5 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array} \quad N' = \begin{array}{c} \begin{matrix} x_2 & x_1 \end{matrix} \\ \begin{bmatrix} 0 & 1 \\ 2 & 0 \\ 1 & 1 \end{bmatrix} \end{array}$$

$$x_{B'} = \begin{bmatrix} x_5 \\ x_3 \\ x_4 \end{bmatrix} \quad x_{N'} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}$$

Example Continued

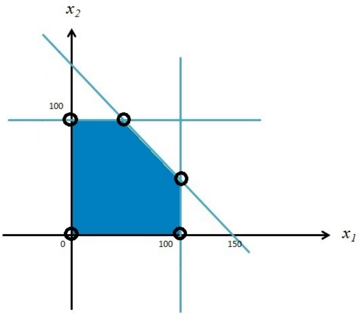
We can list the basic (feasible) solutions

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.



Quiz

$x_N = 0$, $x_B = B^{-1} b$ $x_B \in R^m$

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

It's possible that some $x_B = 0$

- No more than m

Most of time $x_B \neq 0$, $x_B > 0$

- 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

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?

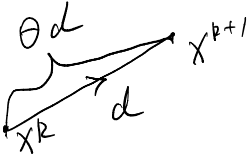
- One may suggest to list all the basic feasible solutions and compare their objective values. However, there are too many of them.
- For a linear optimization with m constraints and n variables, how many basic feasible solutions it may have?
- $C(n, m)$.. If $n = 1000$, $m = 100$, then the value is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

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

Basic Structure of an Optimization Algorithm

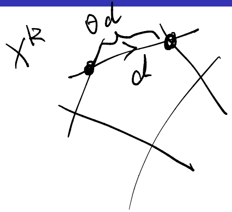
At each iteration k,

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.



$\theta \geq 0$

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

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:

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

Direction Vector Partition

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

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \\ d_1 \\ d_4 \end{bmatrix}, \quad \mathbf{d}_B = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \end{bmatrix} \begin{array}{l} \leftrightarrow x_2 \\ \leftrightarrow x_3 \\ \leftrightarrow x_5 \end{array} \mathbf{d}_N = \begin{bmatrix} d_1 \\ d_4 \end{bmatrix} \begin{array}{l} \leftrightarrow x_1 \\ \leftrightarrow x_4 \end{array}$$

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

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

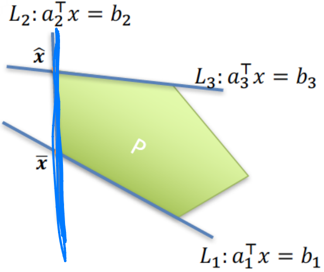
$\hat{x}$ satisfies $A\hat{x} = b$

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

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

Adjacent BFS in Standard Form LP

- 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 differs one element

x_B differs one element

- Two adjacent BFS differ the basis matrix 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.

Adjacent BFS:

- differ one column of B
- differ one column of N
- differ one element in x_B
- differ one element in x_N

x_j $\quad 0 \to$ positive number

$$d_j = 1$$

x_j enters the basis:

x_j goes from nonbasic variable to basic variable

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 $d_j = 1$.

$$\begin{aligned} & Bd_B + Nd_N = 0 \\ \Rightarrow \quad & Bd_B + A_j = 0 \\ \Rightarrow \quad & d_B = -B^{-1}A_j \end{aligned}$$

j^{th} column of A

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**.

Example

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

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

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

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

- 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:

$$d_N = \begin{bmatrix} d_1 \\ d_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{array}{l} \leftrightarrow X_1 \\ \leftrightarrow X_4 \end{array}$$

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

$$d_B = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \end{bmatrix} = -B^{-1} A_4 = - \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \underline{\begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}} \begin{array}{l} \leftrightarrow X_2 \\ \leftrightarrow X_3 \\ \mapsto X_5 \end{array}$$

Outline

Objective Function Partition

The linear objective is:

$$\mathbf{c}^T\mathbf{x} = \begin{bmatrix} \underbrace{c_2 \quad c_3 \quad c_5}_{} & c_1 \quad c_4 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix} = c_1x_1 + \cdots + c_5x_5$$

C_B $\quad$ C_N

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

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix} = \begin{array}{c|c} \begin{bmatrix} c_2 \\ c_3 \\ c_5 \\ c_1 \\ c_4 \end{bmatrix} & \begin{array}{c} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{array} \end{array} \quad \mathbf{c}_B = \begin{bmatrix} c_2 \\ c_3 \\ c_5 \end{bmatrix}, \quad \mathbf{c}_N = \begin{bmatrix} c_1 \\ c_4 \end{bmatrix}$$

$$C = \begin{bmatrix} C_B \\ X_B \end{bmatrix} \qquad C^T X = [C_B^T \; C_N^T] \begin{bmatrix} X_B \\ X_N \end{bmatrix}$$

So the objective becomes:

$$\rightarrow \quad \mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N$$

Choosing a Direction and Pivoting

- 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.
- 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?
  - $c^\top(x + \theta d) - c^\top x$ 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

I want to pick x_j such that my next iteration $\hat{x} = x + \theta d$ is better than current iteration

new objective $-$ old objective < 0

$$= c^T \hat{x} - c^T x$$

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

$$= \underset{\geq 0}{\theta} \underset{< 0}{c^T d} \leftarrow \text{change in objective}$$

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

$$= \theta \begin{bmatrix} c_B^T & c_N^T \end{bmatrix} \begin{bmatrix} -B^{-1} A_j \\ e_j \end{bmatrix}$$

$$= \theta \left[-c_B^T B^{-1} A_j + \underbrace{c_N^T e_j}_{c_j} \right]$$

Cost Change

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

< 0

Reduced Cost

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

- 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.

Optimality Conditions in Simplex

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

- Thus, we want to pick $j \in \cancel{J}$ 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.

Example

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

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

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

$$\bar{c}_1 = \underbrace{-1}_{c_1} - \underbrace{\begin{bmatrix} -2 & 0 & 0 \end{bmatrix}}_{c_B^T} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}}_{A_1} = -1$$

$$\bar{c}_4 = \underbrace{0}_{c_4} - \underbrace{\begin{bmatrix} -2 & 0 & 0 \end{bmatrix}}_{c_B^T} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{A_4} = 1$$

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

Outline

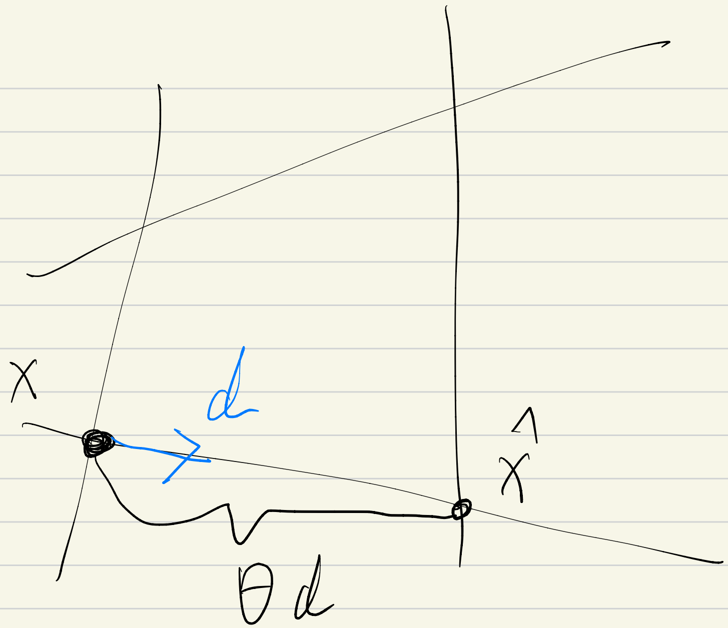
Change of Basis

$$d = \begin{bmatrix} d_B \\ d_N \end{bmatrix} = \begin{bmatrix} -B^{-1}A_j \\ e_j \end{bmatrix}$$

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

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



$\hat{x}$ satisfies $A\hat{x}=b$ ✓

Need $\hat{x} \geq 0$

$x+\theta d \geq 0$

$$\boxed{\begin{aligned} \theta d &\geq -x \\ \theta &\geq -\frac{x}{d} \end{aligned}}$$

Min-ratio Test

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

- 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:

opt problem →

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

↔ min-ratio test

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.

$$x + \theta d \geq 0$$

$$x_B + \theta d_B \geq 0$$

$$\underset{0}{x_N} + \overset{\geq 0}{\theta} \underset{e_j \geq 0}{d_N} \geq 0$$

must satisfy

$$\overset{\geq 0}{x_B} + \overset{\geq 0}{\theta} d_B \geq 0$$

$$\Rightarrow \overset{\geq 0}{x_{B_i}} + \overset{\geq 0}{\theta} d_{B_i} \geq 0$$

$$\Rightarrow \theta \leq -\frac{x_{B_i}}{d_{B_i}} \quad \forall i \in I_B$$

The only possibility that

$$\underset{\geq 0}{x_{B_i}} + \underset{\geq 0}{\theta} \underset{< 0}{d_{B_i}} < 0$$

is $d_{B_i} < 0$

$$\underbrace{x_{B_i}}_{\geq 0} + \underbrace{\theta}_{\geq 0} \underbrace{d_{B_i}}_{< 0} \text{ decreases to } 0$$

for the optimal i in min-ratio test.

We call x_i exits the basis!

x_i goes from basic variable to

non basic variable.

Outline

New solution: j enters and i exits basis

Select x_j, $j \in I_N$ to enter the basis

One x_i, $i \in I_B$ to exit the basis

$k \in I_N$ next iteration

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

$k \in I_B$ x_k

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

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

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

$\bar{c}_j \geq 0$

   - 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 $\boldsymbol{d} = \begin{bmatrix} -B^{-1}A_j \\ e_j \end{bmatrix} \begin{array}{l} \to d_B \\ \to d_N \end{array}$

   $d_B \geq 0$

   - 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.

Remark:

1. After selecting x_j, $j \in I_N$ with $\bar{c}_j < 0$, we can just find the new BFS by

$$\begin{cases} x_B = B^{-1} b \\ x_N = 0 \end{cases}$$

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

$\begin{cases} \bar{c}_3 < 0 \\ \bar{c}_5 > 0 \end{cases}$ $\rightarrow$ Select x_3 to enter the basis

We don't necessary to find d and θ, because we know in the next iteration, You can try $x_N = \begin{bmatrix} x_1 \\ x_5 \end{bmatrix}$ $x_B = \begin{bmatrix} x_3 \\ x_2 \\ x_4 \end{bmatrix}$

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) in Standard form LP

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

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:
  - $\theta^* = \min\left\{\frac{0}{1}, \frac{2}{1}\right\} = 0$, 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.

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, $x_i = 0$.

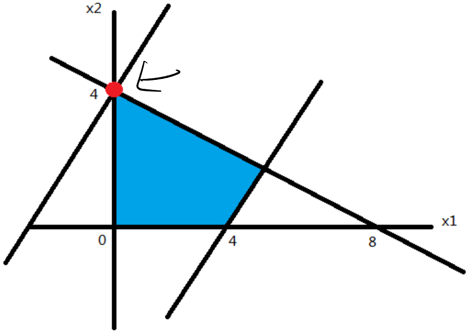
We can still view that we changed the basis from i (i leaving the basis) to j (j entering the basis) and proceed to the next iteration.

- 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

which x_j to enter 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 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

select which x_i to exit basis

Recall that

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

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.

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,

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

Finding an Initial BFS

- One could test different basis B, to see if $A_B^{-1}\boldsymbol{b} \geq 0$.
- 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).

$$\begin{aligned} \text{minimize}_{\boldsymbol{x},\boldsymbol{y}} \quad & \boldsymbol{e}^T \boldsymbol{y} \\ \text{subject to} \quad & A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq 0 \end{aligned}$$

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

There is a trivial BFS to the auxiliary problem: $(\boldsymbol{x} = 0, \boldsymbol{y} = \boldsymbol{b} \geq 0)$ 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.

Two-Phase Simplex Method

By this 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 $(\boldsymbol{x}^*, \mathbf{0})$. Then we know that $\boldsymbol{x}^*$ 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.

Procedure of the Two-Phase Method

Phase I:

1. Construct the auxiliary problem such that $\boldsymbol{b} \geq 0$
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 $\boldsymbol{x}^*$, then we enter phase II

Phase II: Solve the original problem starting from the BFS $\boldsymbol{x}^*$

The Big-M method

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

$$\begin{array}{rc} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} + M\sum_{i=1}^m y_i \\ \text{subject to} & A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq 0 \end{array}$$

This problem has an initial BFS $\boldsymbol{y} = \boldsymbol{b} \geq 0$ (again assuming $\boldsymbol{b} \geq 0$). 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 $\boldsymbol{y}$ must be 0
- Two-Phase is more common