Operations Research III: Theory

Course Overview

Ling-Chieh Kung

Department of Information Management
National Taiwan University

Road map

- **Course overview**.
- Reviewing the simplex method.
- The simplex method in matrices.
- Examples.

Theory

- We have learned models and algorithms.
- In this course, we study **theory**.
- In most cases, these theories are about **optimality conditions**.
  - For example, through observation and analysis we know "for any linear program, if there is an optimal solution, there is an extreme point optimal solution."
  - This allows us to focus only on extreme points (basic feasible solutions).
- In general, **analysis** generate theories, which guide us to develop better algorithms.

Theory

- We have optimality conditions for linear, integer, and nonlinear programming.
- We will introduce a few:
  - Linear programming: duality.
  - Integer programming: total unimodularity.
  - Nonlinear programming: the KKT condition.
- We will also see their applications.
  - Sensitivity analysis.
  - Acceleration for the branch-and-bound algorithm.
  - Network flow models.
  - Linear regression.
  - Support vector machine.

Prerequisites

- We need more solid mathematical foundation.
  - (A lot of) **linear algebra**.
  - (Some) **differential calculus**.
  - (A little bit) discrete mathematics.
- Let's first get familiar with matrices and the matrix notation.
  - In particular, let's review the simplex method and see how it may be executed in a matrix way.

Road map

- Course overview.
- **Reviewing the simplex method**.
- The simplex method in matrices.
- Examples.

An example

- Consider the following linear program

$$
\begin{array}{rrcrcl}
\max & x_1 & & & & \\
\text{s.t.} & 2x_1 & - & x_2 & \leq & 4 \\
 & 2x_1 & + & x_2 & \leq & 8 \\
 & & & x_2 & \leq & 3 \\
 & & & \multicolumn{3}{l}{x_i \geq 0 \quad \forall\ i = 1, 2.}
\end{array}
$$

- The standard form is

$$
\begin{array}{rrcrcrcrcrcl}
\max & x_1 & & & & & & & & & & \\
\text{s.t.} & 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\
 & 2x_1 & + & x_2 & & & + & x_4 & & & = & 8 \\
 & & & x_2 & & & & & + & x_5 & = & 3 \\
 & \multicolumn{11}{l}{x_i \geq 0 \quad \forall\ i = 1, ..., 5.}
\end{array}
$$

The first iteration

- We prepare the initial tableau. We have $x^1 = (0, 0, 4, 8, 3)$ and $z_1 = 0$.

$$
\begin{array}{ccccc|c}
-1 & 0 & 0 & 0 & 0 & 0 \\
\hline
2 & -1 & 1 & 0 & 0 & x_3 = 4 \\
2 & 1 & 0 & 1 & 0 & x_4 = 8 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
$$

- For this **maximization** problem, we look for **negative** numbers in the 0th row. Therefore, x_1 enters.
  - Those numbers in the 0th row are called **reduced costs**.
  - The 0th row is $z - x_1 = 0$. Increasing x_1 can increase z.
- "Dividing the RHS column by the entering column" tells us that x_3 should leave (it has the minimum ratio).[1]
  - This is called the **ratio test**. We **always** look for the smallest ratio.

[1] The 0 in the 3rd row means that increasing x_1 does not affect x_5.

The first iteration

- x_1 enters and x_3 leaves. The next tableau is found by **pivoting** at 2:

$$
\begin{array}{ccccc|c}
-1 & 0 & 0 & 0 & 0 & 0 \\
\hline
\boxed{2} & -1 & 1 & 0 & 0 & x_3 = 4 \\
2 & 1 & 0 & 1 & 0 & x_4 = 8 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
\quad \rightarrow \quad
\begin{array}{ccccc|c}
0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & 2 \\
\hline
1 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & x_1 = 2 \\
0 & 2 & -1 & 1 & 0 & x_4 = 4 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
$$

- The new bfs is $x^2 = (2, 0, 0, 4, 3)$ with $z_2 = 2$.
- Continue?
  - There is a negative reduced cost in the 2nd column: x_2 enters.
- Ratio test:
  - That $-\frac{1}{2}$ in the 1st row shows that increasing x_2 makes x_1 larger. Row 1 does not participate in the ratio test.
  - For rows 2 and 3, row 2 wins (with a smaller ratio).

The second iteration

- x_2 enters and x_4 leaves. We pivot at 2.
- The second iteration is

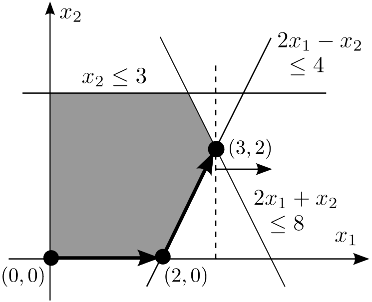
$$
\begin{array}{ccccc|c}
0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & 2 \\
\hline
1 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & x_1 = 2 \\
0 & \boxed{2} & -1 & 1 & 0 & x_4 = 4 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
\quad \rightarrow \quad
\begin{array}{ccccc|c}
0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 3 \\
\hline
1 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & x_1 = 3 \\
0 & 1 & \frac{-1}{2} & \frac{1}{2} & 0 & x_2 = 2 \\
0 & 0 & \frac{1}{2} & \frac{-1}{2} & 1 & x_5 = 1
\end{array}
$$

- The third bfs is $x^3 = (3, 2, 0, 0, 1)$ with $z_3 = 3$.
  - It is optimal (why?).
  - Typically we write the optimal solution we find as x^* and optimal objective value as z^*.

Verifying our solution

- The three basic feasible solutions we obtain are
  - $x^1 = (0, 0, 4, 8, 3)$.
  - $x^2 = (2, 0, 0, 4, 3)$.
  - $x^3 = x^* = (3, 2, 0, 0, 1)$.

  Do they fit our graphical approach?



Road map

- Course overview.
- Reviewing the simplex method.
- **The simplex method in matrices**.
- Examples.

The matrix way for the simplex method

- The easiest way of doing this is to use the **matrix representation** for the simplex method.
- Consider a standard form linear program

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

- By dividing x to x_B and x_N, the basic and nonbasic variables, we have

$$\begin{aligned} \max \quad & c_B^T x_B + c_N^T x_N \\ \text{s.t.} \quad & A_B x_B + A_N x_N = b \\ & x_B, x_N \geq 0. \end{aligned}$$

  - $c^T = [c_B^T, c_N^T]$, $A = [A_B, A_N]$. Note that $A_B \neq I$!

An example

- For our example:

$$
\begin{array}{rrrrrrrrrrrrr}
\max & x_1 & & & & & & & & & & & \\
\text{s.t.} & 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\
 & 2x_1 & + & x_2 & & & + & x_4 & & & = & 8 \\
 & & & x_2 & & & & & + & x_5 & = & 3 \\
\end{array}
$$
$$
x_i \geq 0 \quad \forall i = 1, ..., 5.
$$

- Using the above notation, we have

$$
c^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix},
$$
$$
A = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and } b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}.
$$

An example

- With

$$c^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and } b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix},$$

given $x_B = (x_1, x_4, x_5)$ and $x_N = (x_2, x_3)$, we have

$$c_B^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad c_N^T = \begin{bmatrix} 0 & 0 \end{bmatrix},$$

$$A_B = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}.$$

The matrix way for the simplex method

- Let's rearrange the constraint $A_B x_B + A_N x_N = b$ and obtain

$$\begin{aligned} \max \quad & c_B^T x_B + c_N^T x_N \\ \text{s.t.} \quad & x_B = A_B^{-1}(b - A_N x_N) \\ & x_B, x_N \geq 0. \end{aligned}$$

- Let's replace x_B in the objective function by $A_B^{-1}(b - A_N x_N)$ to

$$\max \; c_B^T [A_B^{-1}(b - A_N x_N)] + c_N^T x_N$$

- The standard form linear program becomes

$$\begin{aligned} \max \quad & c_B^T A_B^{-1} b - (c_B^T A_B^{-1} A_N - c_N^T) x_N \\ \text{s.t.} \quad & x_B = A_B^{-1}(b - A_N x_N) \\ & x_B, x_N \geq 0. \end{aligned}$$

The matrix way for the simplex method

- Finally, let's rearrange the terms in the constraints to obtain

$$
\begin{array}{rl}
\max & c_B^T A_B^{-1} b - (c_B^T A_B^{-1} A_N - c_N^T) x_N \\
\text{s.t.} & I x_B + A_B^{-1} A_N x_N = A_B^{-1} b \\
& x_B, x_N \geq 0.
\end{array}
$$

- Let's ignore the sign constraints and let z be the objective value. We then have

$$
\begin{array}{ccccccc}
z & & + & \left(c_B^T A_B^{-1} A_N - c_N^T\right) x_N & = & c_B^T A_B^{-1} b \\
& I x_B & + & A_B^{-1} A_N x_N & = & A_B^{-1} b.
\end{array}
$$

The matrix way for the simplex method

- Therefore, given **any valid choice of B and N** (the index sets of basic and nonbasic variables), we may use the following table to calculate the simplex tableau:

0	$c_B^T A_B^{-1} A_N - c_N^T$	$c_B^T A_B^{-1} b$	0
I	$A_B^{-1} A_N$	$A_B^{-1} b$	$1, ..., m$
basic	nonbasic	RHS	

- Let's see some examples.

Road map

- Course overview.
- Reviewing the simplex method.
- The simplex method in matrices.
- **Examples**.

An example

- Consider the example again:

$$
\begin{array}{rrrrrrrrrrrr}
\max & x_1 \\
\text{s.t.} & 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\
& 2x_1 & + & x_2 & & & + & x_4 & & & = & 8 \\
& & & x_2 & & & & & + & x_5 & = & 3 \\
& & x_i \geq 0 & \forall i = 1, ..., 5.
\end{array}
$$

- In the matrix representation, we have

$$
c^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix},
$$

$$
A = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and } b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}.
$$

A feasible basis

- Given $x_B = (x_1, x_4, x_5)$ and $x_N = (x_2, x_3)$, we have

$$c_B^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad c_N^T = \begin{bmatrix} 0 & 0 \end{bmatrix},$$
$$A_B = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}.$$

- Given the basis, we have

$$x_B = A_B^{-1} b = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix}, \quad \text{and}$$
$$z = c_B^T A_B^{-1} b = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = 2.$$

- The current bfs is $x = (x_1, x_2, x_3, x_4, x_5) = (2, 0, 0, 4, 3)$.

A feasible basis

- For $x_N = (x_2, x_3)$, the reduced costs are

$$\begin{aligned}\bar{c}_N^T &= c_B^T A_B^{-1} A_N - c_N^T \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \end{bmatrix}.\end{aligned}$$

- x_2 enters. For $x_B = (x_1, x_4, x_5)$, we have
  - $A_B^{-1} A_2 = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ 2 \\ 1 \end{bmatrix}$ and $A_B^{-1} b = \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix}$.
  - $\frac{4}{2} < \frac{3}{1}$, so x_4 leaves.

An optimal basis

- Given $x_B = (x_1, x_2, x_5)$ and $x_N = (x_3, x_4)$ we have

$$c_B^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad c_N^T = \begin{bmatrix} 0 & 0 \end{bmatrix},$$
$$A_B = \begin{bmatrix} 2 & -1 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}.$$

- Given the basis, we have

$$x_B = A_B^{-1} b = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix}, \quad \text{and}$$
$$z = c_B^T A_B^{-1} b = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = 3.$$

- The current bfs is $x = (x_1, x_2, x_3, x_4, x_5) = (3, 2, 0, 0, 1)$.

An optimal basis

- For $x_N = (x_3, x_4)$, the reduced costs are

$$\begin{aligned}\bar{c}_N^T &= c_B^T A_B^{-1} A_N - c_N^T \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \end{bmatrix}.\end{aligned}$$

- No variable should enter: This bfs is optimal.

Another example

- Consider our example:

$$\begin{array}{rrcrcl} \max & 2x_1 & + & 3x_2 & & \\ \text{s.t.} & x_1 & + & x_2 & \leq & 4 \\ & x_1 & + & 2x_2 & \leq & 6 \\ & \multicolumn{5}{l}{x_1, x_2 \geq 0.} \end{array}$$

- Its standard form is

$$\begin{array}{rrcrcrcrcl} \max & 2x_1 & + & 3x_2 & & & & & & \\ \text{s.t.} & x_1 & + & x_2 & + & s_1 & & & = & 4 \\ & x_1 & + & 2x_2 & & & + & s_2 & = & 6 \\ & \multicolumn{9}{l}{x_1, x_2, s_1, s_2 \geq 0.} \end{array}$$

Another example

- Let's say $x_B = (s_1, s_2)$ and $x_N = (x_1, x_2)$.
  - We have $c_B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $c_N = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$, $A_B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $A_N = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, and $b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$.
  - We then have $A_B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $A_B^{-1} A_N = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, $A_B^{-1} b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$, $c_B^T A_B^{-1} A_N - c_N^T = \begin{bmatrix} -2 & -3 \end{bmatrix}$, and $c_B^T A_B^{-1} b = 0$.
  - This gives us exactly the initial tableau

$$
\begin{array}{cccc|c}
-2 & -3 & 0 & 0 & 0 \\
\hline
1 & 1 & 1 & 0 & 4 \\
\underbrace{1 \quad\quad 2}_{\text{nonbasic}} & & \underbrace{0 \quad 1}_{\text{basic}} & & \underbrace{6}_{\text{RHS}}
\end{array}
$$

Another example

- Let's say $x_B = (x_1, x_2)$ and $x_N = (s_1, s_2)$.
  - We have $c_B = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$, $c_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $A_B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, $A_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$.
  - We then have $A_B^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$, $A_B^{-1}A_N = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$, $A_B^{-1}b = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, $c_B^T A_B^{-1} A_N - c_N^T = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and $c_B^T A_B^{-1} b = 10$.
  - This gives us exactly the optimal tableau

$$
\begin{array}{cccc|c}
0 & 0 & 1 & 1 & 10 \\
\hline
1 & 0 & -2 & -1 & 2 \\
\underbrace{0 \quad 1}_{\text{basic}} & & \underbrace{-1 \quad 1}_{\text{nonbasic}} & & \underbrace{2}_{\text{RHS}}
\end{array}
$$

The matrix way

- In short, the simplex method may be executed with matrix calculations.
  - It is a process of searching for an optimal basis.
  - In each iteration, we replace a basic variable by a nonbasic one.
- In this way, the **bottleneck** is the calculation of A_B^{-1}.
  - This explains why the execution time of the simplex method is usually proportional to m^3, where m is the number of constraints.
- Actually we may be faster.
  - Because the current basis B and the previous one have only **one variable** different, the current A_B and the previous one have only **one column** different.
  - Calculating A_B^{-1} can be faster with the previous inverse.
- In fact, how do you know that A_B is still **invertible** after changing one column?