Convex Optimization

Lecture 4: Two-phase Simplex, Degenerated form and Duality of LP

Lecturer: *Dr*. Wan-Lei Zhao

Autumn Semester 2025

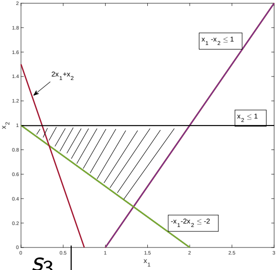
Contact: wlzhao@xmu.edu.cn

Outline

Linear Programming: the two-phase problem (1)

$$
\text{Max. } 2x_1 + x_2
$$

$$
\text{s. t.} \begin{cases} x_1 - x_2 \leq 1 \\ x_1 + 2x_2 \geq 2 \\ x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases} \quad (1)
$$



	z	x_1	x_2	s_1	s_2	s_3	
	1	-2	-1	0	0	0	0
s_1	0	1	-1	1	0	0	1
s_2	0	1	2	0	-1	0	2
s_3	0	0	1	0	0	1	1

- $x_1 = x_2 = 0$ is not a basic solution for the problem
- There is no start point

Linear Programming: the two-phase problem (2)

- Introduce slack variable s_1, s_2 and s_3

$$\text{Max. } 2x_1 + x_2$$
$$\text{s.t. } \begin{cases} x_1 - x_2 + s_1 = 1 \\ x_1 + 2x_2 - s_2 = 2 \\ x_2 + s_3 = 1 \\ x_1, x_2, s_1, s_2, s_3 \geq 0 \end{cases}$$

- Let $x_1 = x_2 = 0$, we check whether we can have a basic solution
- The answer is No since $s_2 = -2$

- Introduce artificial/auxiliary variable a_1

$$\text{Max. } 2x_1 + x_2$$
$$\text{s.t. } \begin{cases} x_1 - x_2 + s_1 = 1 \\ x_1 + 2x_2 - s_2 + a_1 = 2 \\ x_2 + s_3 = 1 \\ x_1, x_2, s_1, s_2, s_3, a_1 \geq 0 \end{cases}$$

- Let $x_1 = x_2 = 0$, we check whether we can have a basic solution
- The answer is Yes since $a_1 = 2$

Linear Programming: the two-phase problem (3)

- We solve the auxiliary problem first
- Find out the minimum a_1 that satisfies the equations

$$\text{Max.} \quad -a_1$$

$$\text{s.t.} \quad \begin{cases} x_1 - x_2 + s_1 = 1 \\ x_1 + 2x_2 - s_2 + a_1 = 2 \\ x_2 + s_3 = 1 \\ x_1, x_2, s_1, s_2, s_3, a_1 \geq 0 \end{cases}$$

	B	z	a_1	x_1	x_2	s_1	s_2	s_3	C
R_0		1	1	0	0	0	0	0	0
R_1	s_1	0	0	1	-1	1	0	0	1
R_2	a_1	0	1	1	2	0	-1	0	2
R_3	s_3	0	0	0	1	0	0	1	1

Linear Programming: the two-phase problem (4)

Phase-I: pre-processing

	B	z	a_1	x_1	x_2	s_1	s_2	s_3	C
R_0		1	1	0	0	0	0	0	0
R_1	s_1	0	0	1	-1	1	0	0	1
R_2	a_1	0	1	1	2	0	-1	0	2
R_3	s_3	0	0	0	1	0	0	1	1

$$R_0 = R_0 - R_2 \Downarrow$$

	B	z	a_1	x_1	x_2	s_1	s_2	s_3	C
R_0		1	0	-1	-2	0	2	0	-2
R_1	s_1	0	0	1	-1	1	0	0	1
R_2	a_1	0	1	1	2	0	-1	0	2
R_3	s_3	0	0	0	1	0	0	1	1

Pre-processing: to make sure the coefficients of all basic variables on R_0 are zeros

Linear Programming: the two-phase problem (5)

	B	z	a_1	x_1	x_2	s_1	s_2	s_3	C
R_0		1	0	-1	-2	0	2	0	-2
R_1	s_1	0	0	1	-1	1	0	0	1
R_2	a_1	0	1	1	2	0	-1	0	2
R_3	s_3	0	0	0	1	0	0	1	1

swap x_2 in $\Downarrow$ swap a_1 out

	B	z	a_1	x_1	x_2	s_1	s_2	s_3	C
R_0		1	1	0	0	0	0	0	0
R_1	s_1	0	0.5	1.5	0	1	-0.5	0	2
R_2	x_2	0	1	1	2	0	-1	0	2
R_3	s_3	0	-0.5	-0.5	0	0	0.5	1	0

- To this end, this auxiliary problem reaches the optima
- We will solve the original problem with this tableau
- Only need to replace the objective function

Linear Programming: the two-phase problem (6)

Phase-II

	B	z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	-2	-1	0	0	0	0
R_1	s_1	0	1.5	0	1	-0.5	0	2
R_2	x_2	0	1	2	0	-1	0	2
R_3	s_3	0	-0.5	0	0	0.5	1	0

$$R_0 = R_0 + R_2/2 \Downarrow$$

	B	z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	-1.5	0	0	-0.5	0	1
R_1	s_1	0	1.5	0	1	-0.5	0	2
R_2	x_2	0	1	2	0	-1	0	2
R_3	s_3	0	-0.5	0	0	0.5	1	0

- At Phase-II, the auxiliarly variables like a_1 can be simply ignored!!

Pre-processing: to make sure the coefficients of all basic variables on R_0 are zeros

Linear Programming: the two-phase problem (7)

Phase-II

B		z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	-1.5	0	0	-0.5	0	1
R_1	s_1	0	1.5	0	1	-0.5	0	2
R_2	x_2	0	1	2	0	-1	0	2
R_3	s_3	0	-0.5	0	0	0.5	1	0

swap-in x_1 $\Downarrow$ swap-out s_1

B		z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	0	0	1	-1	0	3
R_1	x_1	0	1.5	0	1	-0.5	0	2
R_2	x_2	0	0	2	-2/3	-2/3	0	2/3
R_3	s_3	0	0	0	1/3	1/3	1	2/3

Linear Programming: the two-phase problem (8)

Phase-II

B		z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	0	0	1	-1	0	3
R_1	x_1	0	1.5	0	1	-0.5	0	2
R_2	x_2	0	0	2	-2/3	-2/3	0	2/3
R_3	s_3	0	0	0	1/3	1/3	1	2/3

swap-in s_2 $\Downarrow$ swap-out s_3

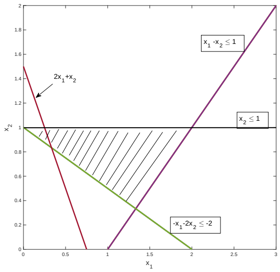
B		z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	0	0	2	0	3	5
R_1	x_1	0	1.5	0	1.5	0	1.5	3
R_2	x_2	0	0	2	0	0	2	2
R_3	s_2	0	0	0	1/3	1/3	1	2/3

Linear Programming: the two-phase problem (9)

	B	z	x_1	x_2	s_1	s_2	s_3	C
R_0		1	0	0	2	0	3	5
R_1	x_1	0	1.5	0	1.5	0	1.5	3
R_2	x_2	0	0	2	0	0	2	2
R_3	s_2	0	0	0	1/3	1/3	1	2/3

- We know that $x_1 = 2, x_2 = 1, s_2 = 2, s_1 = s_3 = 0$

$$\text{Max. } 2x_1 + x_2$$
$$\text{s. t.} \begin{cases} x_1 - x_2 \leq 1 \\ x_1 + 2x_2 \geq 2 \\ x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases} \quad (1)$$



Supporting Theorem for Two-Phase Method

- Given the original problem is **P**, the auxiliary is **A**

$$\mathbf{P}\text{: Max. } C^T X$$

$$\text{s. t.} \begin{cases} e_{11}x_1 + \cdots + e_{1n}x_n = b_1 \\ e_{21}x_1 + \cdots + e_{2n}x_n = b_2 \\ \cdots \\ e_{m1}x_1 + \cdots + e_{mn}x_n = b_m \\ x_{1\cdots n} \geq 0 \end{cases}$$

$$\mathbf{A}\text{: Max. } -\sum_{i=1}^{m} a_i$$

$$\text{s. t.} \begin{cases} e_{11}x_1 + \cdots + e_{1n}x_n + a_1 = b_1 \\ e_{21}x_1 + \cdots + e_{2n}x_n + a_2 = b_2 \\ \cdots \\ e_{m1}x_1 + \cdots + e_{mn}x_n + a_m = b_m \\ x_{1\cdots n} \geq 0, a_{1\cdots m} \geq 0 \end{cases}$$

Theorem: **P** has a feasible solution $\Leftrightarrow$ **A** has optimal value 0.

Proof for the Supporting Theorem

- **P** has a feasible solution $\Longrightarrow$ **A** has optimal value 0.
- Given the feasible solution for **P** is $[x_1^*, x_2^*, \cdots, x_n^*]^T$
- $\Longrightarrow [x_1^*, x_2^*, \cdots, x_n^*, 0, \cdots, 0]^T$ is feasible for **A**
- Moreover, $[x_1^*, x_2^*, \cdots, x_n^*, 0, \cdots, 0]^T$ is optimal for **A** with value 0

- **P** has a feasible solution $\Longleftarrow$ **A** has optimal value 0.
- Given the optimal solution for **A** is $[x_1^*, x_2^*, \cdots, x_n^*, a_1^*, \cdots, a_k^*]^T$ with value 0
- $\Longrightarrow [a_1^*, \cdots, a_k^*] = [0, \cdots, 0]^T$
- $\Longrightarrow [x_1^*, x_2^*, \cdots, x_n^*]^T$ is feasible for **P**

— **proved** —

Linear Programming: the two-phase problem (9)

- Summary
  - When there is no basic solution
  - LP problem should be addressed in two phases
  - **Phase-I:** 1. Introduce artificial variables; 2. Solve an auxiliary problem
  - **Phase-II:** 1. Replace to original objective; 2. Solve the original problem
  - Keep the basic variables to 0 in the objective

Outline

Two-phase Simplex by Matrix Operation (1)

- Similar as before, in order to address LP in large-scale
- We should do Simplex with matrix operations
- In the following, we will formulate the Simplex method in matrix operations

Two-phase Simplex by Matrix Operation (2)

- We now consider how to solve LP in two phases
- Let's consider the first phase problem
- $C_b = [0\ 1\ 0]$, $C_n = [1\ 0\ 0]$

$$z + a_1 = 0$$

$$\text{s.t.} \begin{cases} x_1 - x_2 + s_1 \qquad = 1 \\ -x_1 - 2x_2 + s_2 \ - a_1 = -2 \\ \qquad x_2 + \quad s_3 = 1 \\ a_1, x_1, x_2, s_1, s_2, s_3 \geq 0 \end{cases} \tag{2}$$

$$B = \begin{array}{c} \begin{matrix} s_1 & a_1 & s_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad N = \begin{array}{c} \begin{matrix} x_1 & x_2 & s_2 \end{matrix} \\ \begin{bmatrix} 1 & -1 & 0 \\ -1 & -2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \qquad b = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Two-phase Simplex by Matrix Operation: Phase-1 (1)

- $C_b = C_b + \sum_i a_i =$, $C_n = [-1\ 0\ 0]$

$$B = \begin{array}{c} \begin{array}{ccc} s_1 & a_1 & s_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \quad N = \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & s_2 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 \\ -1 & -2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \quad b = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

1. Let $k = 1$, $p_k = N(:, k)$, calculate $y_k = B^{-1} \cdot p_k$, $b_1 = B^{-1} \cdot b$
2. $\frac{b_1}{y_k} = \{\frac{1}{-1}, \frac{-2}{-1}, \frac{1}{-1}\}$, $r = 2$, we get s_2
3. Swap-in x_0 to **B**, swap-out s_2 to **N**
4. $C_b = [0\ -1\ 0]$, $C_n = [0\ 0\ 0]$

$$B = \begin{array}{c} \begin{array}{ccc} s_1 & x_0 & s_3 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \end{array} \quad B^{-1} = \begin{array}{c} \begin{array}{ccc} s_1 & x_0 & s_3 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \end{array} \quad N = \begin{array}{c} \begin{array}{ccc} s_2 & x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Summary over Two-phase Simplex

A. Stage-1: Solve the auxiliary problem
   1. Set k=1, Repeat
   2. Find out r by $\arg\min_r\{B^{-1}\cdot b./B^{-1}\cdot N(:,k)\}$
   3. Swap-in $N(:,k)$ to B, Swap-out $B(:,r)$ to N_2
   4. $Cn(k) \rightleftharpoons Cb(r)$
   5. Find out k by $\arg\min_k\{C_b\cdot B^{-1}N - C_n\}$

B. Stage-2: Solve the original problem
   1. Set $C_b = [0 \ \cdots \ 0]$, $C_n = -C\cdot N\cdot B^{-1}$
   2. $B_2 = N, N_2 = B^{-1}$, $b = B^{-1}\cdot b$
   3. Repeat
   4. Find out k by $\arg\min_k\{C_b\cdot B_2^{-1}N_2 - C_n\}$
   5. Find out r by $\arg\min_r\{B_2^{-1}\cdot b./B_2^{-1}\cdot N_2(:,k)\}$
   6. Swap-in $N_2(:,k)$ to B_2, Swap-out $B_2(:,r)$ to N_2
   7. $Cn(k) \rightleftharpoons Cb(r)$

Outline

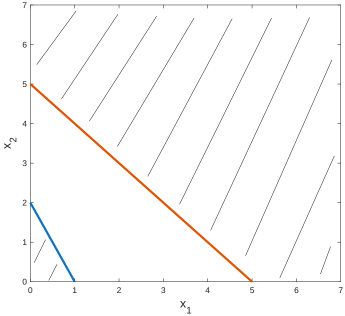
Infeasible Linear Programming Problem (1)

- In practice, not all the LP problems are feasible
- When the feasible set is empty, it becomes infeasible

Infeasible Linear Programming Problem (2)

- **An LP is infeasible if there is no feasible solution**

$$
\begin{aligned}
&\text{Max. } 3x_1 + x_2 \\
&\text{s.t. } \begin{cases} 2x_1 + x_2 \leq 2 \\ x_1 + x_2 \geq 5 \\ x_1, x_2 \geq 0 \end{cases}
\end{aligned} \tag{3}
$$



Infeasible Linear Programming Problem (3)

- **An LP is infeasible if there is no feasible solution**

- Introduce slack variable s_1 and s_2

$$\text{Max. } 3x_1 + x_2$$
$$\text{s.t. } \begin{cases} 2x_1 + x_2 + s_1 = 2 \\ x_1 + x_2 - s_2 = 5 \\ x_1, x_2, s_1, s_2 \geq 0 \end{cases}$$

- Introduce artificial/auxiliary variable a_1

$$\text{Max. } 3x_1 + x_2$$
$$\text{s.t. } \begin{cases} 2x_1 + x_2 + s_1 \qquad = 2 \\ x_1 + x_2 - s_2 + a_1 = 5 \\ x_1, x_2, s_1, s_2, a_1 \geq 0 \end{cases}$$

Infeasible Linear Programming Problem (3)

$$\text{Max. } 3x_1 + x_2$$
$$\text{s.t.} \begin{cases} 2x_1 + x_2 + s_1 = 2 \\ x_1 + x_2 - s_2 + a_1 = 5 \\ x_1, x_2, s_1, s_2, a_1 \geq 0 \end{cases}$$

- Solve the auxiliary problem first

$$\text{Max. } -a_1$$
$$\text{s.t.} \begin{cases} 2x_1 + x_2 + s_1 = 2 \\ x_1 + x_2 - s_2 + a_1 = 5 \\ x_1, x_2, s_1, s_2, a_1 \geq 0 \end{cases}$$

Phase-I: pre-processing

	B	z	a_1	x_1	x_2	s_1	s_2	C
R_0		1	1	0	0	0	0	0
R_1	s_1	0	0	2	1	1	0	2
R_2	a_1	0	1	1	1	0	-1	5

$$R_0 = R_0 - R_2 \Downarrow$$

	B	z	a_1	x_1	x_2	s_1	s_2	C
R_0		1	0	-1	-1	0	1	-5
R_1	s_1	0	0	2	1	1	0	2
R_2	a_1	0	1	1	1	0	-1	5

Infeasible Linear Programming Problem (4)

Phase-I: Round-1

	B	z	a_1	x_1	x_2	s_1	s_2	C
R_0		1	0	-1	-1	0	1	-5
R_1	s_1	0	0	2	1	1	0	2
R_2	a_1	0	1	1	1	0	-1	5

$\Downarrow$

	B	z	a_1	x_1	x_2	s_1	s_2	C
R_0		1	0	0	0	-0.5	0.5	-4
R_1	x_1	0	0	1	1	1	0	2
R_2	a_1	0	1	0	0.5	-0.5	-1	4

$\Downarrow$

	B	z	a_1	x_1	x_2	s_1	s_2	C
R_0		1	0	1	0	1	1	-3
R_1	x_1	0	0	2	1	1	0	2
R_2	a_1	0	1	-1	0	-1	-1	3

Infeasible Linear Programming Problem (5)

	B	z	a_1	x_1	x_2	s_1	s_2	C
R_0		1	0	1	0	1	1	-3
R_1	x_1	0	0	2	1	1	0	2
R_2	a_1	0	1	-1	0	-1	-1	3

$$\text{Max.} \quad -a_1$$

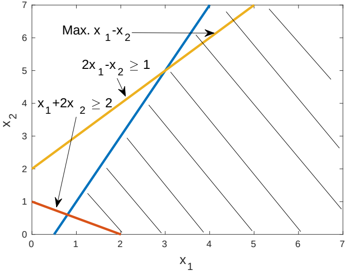
$$\text{s.t.} \begin{cases} 2x_1 + x_2 + s_1 \qquad = 2 \\ x_1 + x_2 - s_2 + a_1 = 5 \\ x_1, x_2, s_1, s_2, a_1 \geq 0 \end{cases}$$

- **Phase-I**: we cannot reach the optima for the auxilary problem
- There is no way to optimize it further
- It becomes the criteria for us to judge whether a problem is feasible

Unbounded Linear Programming Problem (1)

- In some cases, the maximum/minimum value could be $\infty / -\infty$
- In such cases, the LP problems are unbounded

$$\begin{aligned} & \text{Max. } x_1 - x_2 \\ \text{s.t.} & \begin{cases} 2x_1 - x_2 \geq 1 \\ x_1 + 2x_2 \geq 2 \\ x_1, x_2 \geq 0 \end{cases} \end{aligned} \tag{4}$$



Unbounded Linear Programming Problem (2)

- **An LP is unbounded if its feasible solution is infinity**

- Introduce slack variable s_1 and s_2

$$\text{Max. } x_1 - x_2$$
$$\text{s.t. } \begin{cases} 2x_1 - x_2 - s_1 = 1 \\ x_1 + 2x_2 - s_2 = 2 \\ x_1, x_2, s_1, s_2 \geq 0 \end{cases}$$

- Introduce artificial/auxiliary variable a_1 and a_2

$$\text{Max. } x_1 - x_2$$
$$\text{s.t. } \begin{cases} 2x_1 - x_2 - s_1 + a_1 = 1 \\ x_1 + 2x_2 - s_2 + a_2 = 2 \\ x_1, x_2, s_1, s_2, a_1, a_2 \geq 0 \end{cases}$$

Unbounded Linear Programming Problem (2)

- Solve the auxiliary problem first

$$\text{Max. } x_1 - x_2$$

$$\text{s.t. } \begin{cases} 2x_1 - x_2 - s_1 + a_1 = 1 \\ x_1 + 2x_2 - s_2 + a_2 = 2 \\ x_1, x_2, s_1, s_2, a_1, a_2 \geq 0 \end{cases}$$

$$\text{Max. } -a_1 - a_2$$

$$\text{s.t. } \begin{cases} 2x_1 - x_2 - s_1 + a_1 = 1 \\ x_1 + 2x_2 - s_2 + a_2 = 2 \\ x_1, x_2, s_1, s_2, a_1, a_2 \geq 0 \end{cases}$$

Phase-I: pre-processing

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1	1	0	0	0	0	0
a_1	0	1	0	2	-1	-1	0	1
a_2	0	0	1	1	2	0	-1	2

$$R_0 = R_0 - R_1 - R_2 \Downarrow$$

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	0	0	-3	-1	1	1	-3
a_1	0	1	0	2	-1	-1	0	1
a_2	0	0	1	1	2	0	-1	2

Unbounded Linear Programming Problem (3)

- **Phase-I**

Phase-I: Round-1

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	0	0	-3	-1	1	1	-3
a_1	0	1	0	2	-1	-1	0	1
a_2	0	0	1	1	2	0	-1	2

$$\Downarrow$$

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1.5	0	0	-2.5	-0.5	1	-1.5
x_1	0	1	0	2	-1	-1	0	1
a_2	0	-0.5	1	0	2.5	0.5	-1	1.5

Unbounded Linear Programming Problem (4)

Phase-I: Round-2

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1.5	0	0	-2.5	-0.5	1	-1.5
x_1	0	1	0	2	-1	-1	0	1
a_2	0	-0.5	1	0	2.5	0.5	-1	1.5

$$\Downarrow$$

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1	1	0	0	0	0	0
x_1	0	0.8	0.4	2	0	-0.8	-0.4	1.6
x_2	0	-0.5	1	0	2.5	0.5	-1	1.5

Unbounded Linear Programming Problem (5)

$$\text{Max. } x_1 - x_2$$

$$\text{s.t.} \begin{cases} 2x_1 - x_2 - s_1 + a_1 = 1 \\ x_1 + 2x_2 - s_2 + a_2 = 2 \\ x_1, x_2, s_1, s_2, a_1, a_2 \geq 0 \end{cases}$$

Phase-II: replace R_0

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1	1	0	0	0	0	0
x_1	0	0.8	0.4	2	0	-0.8	-0.4	1.6
x_2	0	-0.5	1	0	2.5	0.5	-1	1.5

$$\Downarrow$$

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1	1	-1	1	0	0	0
x_1	0	0.8	0.4	2	0	-0.8	-0.4	1.6
x_2	0	-0.5	1	0	2.5	0.5	-1	1.5

Unbounded Linear Programming Problem (6)

Phase-II: Pre-processing

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	1	1	-1	1	0	0	0
x_1	0	0.8	0.4	2	0	-0.8	-0.4	1.6
x_2	0	-0.5	1	0	2.5	0.5	-1	1.5

$$R_0 = R_0 + R_1/2 - R_2/2.5 \Downarrow$$

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	0.6	-0.2	0	0	-0.6	0.2	0.2
x_1	0	0.8	0.4	2	0	-0.8	-0.4	1.6
x_2	0	-0.5	1	0	2.5	0.5	-1	1.5

Unbounded Linear Programming Problem (7)

Phase-II: Round-1

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	0.6	-0.2	0	0	-0.6	0.2	0.2
x_1	0	0.8	0.4	2	0	-0.8	-0.4	1.6
x_2	0	-0.5	1	0	2.5	0.5	-1	1.5

$\Downarrow$

B	z	a_1	a_2	x_1	x_2	s_1	s_2	C
	1	0	1	0	3	0	-1	2
x_1	0	0	2	2	4	0	-2	4
s_1	0	-0.5	1	0	2.5	0.5	-1	1.5

- To this end, we see this problem is not bounded

Theorem about the Feasibility of LP

- For an arbitrary LP problem in standard form, the following statements hold
  1. If there is no optimal solution, then the problem is either infeasible or unbounded
  2. If a feasible solution exists, then a basic feasible solution exists

Outline

Degenerated Linear Programming Problem (1)

- In our previous lectures, we presented the Simplex method
- Essentially, we jump from one vertex to another to find out the extreme
- The jumping (or swap-in and swap-out operation) always makes the objective function better
- However, this is not necessarily true

Degenerated Linear Programming Problem (2)

- **An LP is degenerate if in a basic feasible solution, one of the basic variables takes on a zero value**

$$\text{Max. } x_1 + x_2 + x_3$$
$$\text{s.t. } \begin{cases} x_1 + x_2 \leq 1 \\ -x_2 + x_3 \leq 0 \\ x_1, x_2 \geq 0 \end{cases} \tag{5}$$

	z	x_1	x_2	x_3	s_1	s_2	
	1	-1	-1	-1	0	0	0
s_1	0	1	1	0	1	0	1
s_2	0	0	-1	1	0	1	0

- $s_1 = 1, s_2 = 0$

Degenerated Linear Programming Problem (3)

- **An LP is degenerate if in a basic feasible solution, one of the basic variables takes on a zero value**

	z	x_1	x_2	x_3	s_1	s_2	
	1	-1	-1	-1	0	0	0
s_1	0	1	1	0	1	0	1
s_2	0	0	-1	1	0	1	0

$$\Downarrow$$

	z	x_1	x_2	x_3	s_1	s_2	
	1	0	0	-1	1	0	1
x_1	0	1	1	0	1	0	1
s_2	0	0	-1	1	0	1	0

- $x_1 = 1, s_2 = 0$

Degenerated Linear Programming Problem (4)

- Degeneracy happens

	z	x_1	x_2	x_3	s_1	s_2	
	1	0	0	-1	1	0	1
x_1	0	1	1	0	1	0	1
s_2	0	0	-1	1	0	1	0

$$\Downarrow$$

	z	x_1	x_2	x_3	s_1	s_2	
	1	0	-1	0	1	1	1
x_1	0	1	1	0	1	0	1
x_3	0	0	-1	1	0	1	0

- $x_1 = 1, x_3 = 0$

Degenerated Linear Programming Problem (5)

- Converges

	z	x_1	x_2	x_3	s_1	s_2	
	1	0	-1	0	1	1	1
x_1	0	1	1	0	1	0	1
x_3	0	0	-1	1	0	1	0

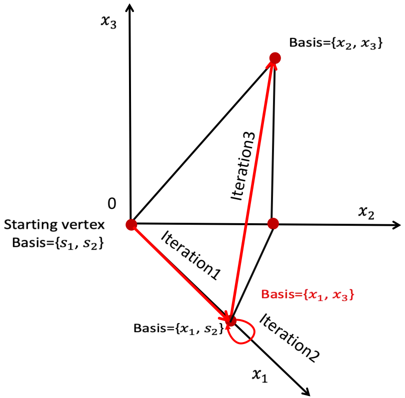
$$\Downarrow$$

	z	x_1	x_2	x_3	s_1	s_2	
	1	1	0	0	2	1	2
x_2	0	1	1	0	1	0	1
x_3	0	1	0	1	1	1	1

- $x_2 = 1, x_3 = 1$

Degenerated Linear Programming Problem (6)

- In practice, there are several ways to address the degeneracy



$$(s_1 = 1, s_2 = 0) \rightarrow (x_1 = 1, s_2 = 0) \rightarrow (x_1 = 1, x_3 = 0) \rightarrow (x_2 = 1, x_3 = 1)$$

Degenerated Linear Programming Problem (7)

- When the degeneracy lasts for a few steps, and the iteration goes back to previous basis
- They cycling happens
- Rules to avoid cycling
  1. **Rule-1:** Choose the variable with the most negative coefficient to enter-in
  2. **Rule-2:** When there are more than one entering or leaving variables, select the one with the smallest index

- **Rule-2** is also known as **Bland's rule**
- The order is defined as $a_1, \cdots, a_k, x_1, \cdots, x_n, s_1, \cdots, s_m$, from now on

Outline

Duality of Linear Programming (1)

- Given the following problem

$$\text{Max. } 7x_1 + 4x_2$$
$$\text{s.t.} \begin{cases} 2x_1 + x_2 \leq 20 \\ x_1 + x_2 \leq 18 \\ x_1 \leq 8 \\ x_1, x_2 \geq 0 \end{cases} \tag{1}$$

- Given $y_1 \geq 0$, $y_2 \geq 0$, $y_3 \geq 0$, the following holds

$$\text{Max. } 7x_1 + 4x_2$$
$$\text{s.t.} \begin{cases} y_1(2x_1 + x_2) \leq 20y_1 \\ y_2(x_1 + x_2) \leq 18y_2 \\ y_3x_1 \leq 8y_3 \\ x_1, x_2 \geq 0 \end{cases} \tag{2}$$

Duality of Linear Programming (2)

$$\text{Max. } 7x_1 + 4x_2$$
$$\text{s.t.} \begin{cases} y_1(2x_1 + x_2) \leq 20y_1 \\ y_2(x_1 + x_2) \leq 18y_2 \\ y_3x_1 \leq 8y_3 \\ x_1, x_2 \geq 0 \end{cases} \tag{2}$$

- So we have

$$(2y_1 + y_2 + y_3)x_1 + (y_1 + y_2)x_2 \leq 20y_1 + 18y_2 + 8y_3 \tag{3}$$

Duality of Linear Programming (3)

$$\text{Max. } 7x_1 + 4x_2$$
$$\text{s.t.} \begin{cases} 2x_1 + x_2 \leq 20 \\ x_1 + x_2 \leq 18 \\ x_1 \leq 8 \\ x_1, x_2 \geq 0 \end{cases} \tag{1}$$

- Since we have
$$(2y_1 + y_2 + y_3)x_1 + (y_1 + y_2)x_2 \leq 20y_1 + 18y_2 + 8y_3 \tag{3}$$

- Let's define another problem

$$\text{Min. } 20y_1 + 18y_2 + 8y_3$$
$$\text{s.t.} \begin{cases} 2y_1 + y_2 + y_3 \geq 7 \\ y_1 + y_2 \geq 4 \\ y_1, y_2, y_3 \geq 0 \end{cases} \tag{4}$$

- Problems in Eqn. 4 and Eqn. 1 are the dual problem of each other

Duality of Linear Programming (4)

$$\text{Max. } 7x_1 + 4x_2$$
$$\text{s.t.} \begin{cases} 2x_1 + x_2 \leq 20 \\ x_1 + x_2 \leq 18 \\ x_1 \leq 8 \\ x_1, x_2 \geq 0 \end{cases} \tag{1}$$

- Solving above problem, we have
- The maximum is 78 when $x_1 = 2, x_2 = 16$

$$\text{Min. } 20y_1 + 18y_2 + 8y_3$$
$$\text{s.t.} \begin{cases} 2y_1 + y_2 + y_3 \geq 7 \\ y_1 + y_2 \geq 4 \\ y_1, y_2, y_3 \geq 0 \end{cases} \tag{4}$$

- Solving above problem, we have
- The minimum is 78 when $y_1 = 3, y_2 = 1, y_3 = 0$

- They share the same extreme value, which can be revealed by

$$(2y_1 + y_2 + y_3)x_1 + (y_1 + y_2)x_2 \leq 20y_1 + 18y_2 + 8y_3 \tag{3}$$

Weak Duality Theorem (1)

- The primal problem

$$\text{Max.} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, \cdots, m$$

$$x_j \geq 0, j = 1, \cdots, n$$

- The dual problem

$$\text{Min.} \quad \sum_{i=1}^{m} b_i y_i$$

$$\text{s.t.} \sum_{i=1}^{m} y_i a_{ij} \geq c_j, j = 1, \cdots, n$$

$$y_i \geq 0, i = 1, \cdots, m$$

- **If $(x_1, \cdots, x_n)$ is feasible for the primal and $(y_1, \cdots, y_m)$ is feasible for the dual, then**

$$\sum_j c_j x_j \leq \sum_i b_i y_i. \tag{5}$$

Weak Duality Theorem (2)

$$\sum_j c_j x_j \leq \sum_i b_i y_i. \tag{5}$$

- **Proof:** since $\sum_{i=1}^{m} y_i a_{ij} \geq c_j, j = 1, \cdots, n$ and $x_j \geq 0$

$$\begin{aligned}
\sum_j c_j x_j &\leq \sum_{j=1}^{n} (\sum_{i=1}^{m} y_i a_{ij}) x_j \\
&= \sum_{ij} y_i a_{ij} x_j \\
&= x_1 \sum_{i=1}^{m} y_i a_{i1} + \cdots + x_n \sum_{i=1}^{m} y_i a_{in} \\
&= \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j) y_i \\
\xrightarrow{\sum_{j=1}^{n} a_{ij} x_j \leq b_i} &\leq \sum_i b_i y_i
\end{aligned}$$

Strong Duality Theorem

- The primal problem

$$\text{Max. } \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, \cdots, m$$

$$x_j \geq 0, j = 1, \cdots, n$$

- The dual problem

$$\text{Min. } \sum_{i=1}^{m} b_i y_i$$

$$\text{s.t.} \sum_{i=1}^{m} y_i a_{ij} \geq c_j, j = 1, \cdots, n$$

$$y_i \geq 0, i = 1, \cdots, m$$

- **If** $x^* = (x_1^*, \cdots, x_n^*)$ **is the optimal solution for the primal and** $y^* = (y_1^*, \cdots, y_m^*)$ **is the optimal solution for the dual, then**

$$\sum_j c_j x_j^* = \sum_i b_i y_i^*. \tag{6}$$

Solve LP by its Dual (1)

- Given following problem

$$\text{Max.} \quad -x_1 - 3x_2$$
$$\text{s.t.} \begin{cases} x_1 + x_2 \geq 4 \\ -4x_1 + x_2 \leq -1 \\ x_2 \geq 3 \\ x_1, x_2 \geq 0 \end{cases} \qquad (7)$$

- Reorganize it into following form

$$\text{Min.} \quad x_1 + 3x_2$$
$$\text{s.t.} \begin{cases} x_1 + x_2 \geq 4 \\ 4x_1 - x_2 \geq 1 \\ x_2 \geq 3 \\ x_1, x_2 \geq 0 \end{cases} \qquad (8)$$

Solve LP by its Dual (2)

- Given following problem

$$\text{Min. } x_1 + 3x_2$$
$$\text{s.t.} \begin{cases} x_1 + 3x_2 \geq 4 \\ 4x_1 - x_2 \geq 1 \\ x_2 \geq 3 \\ x_1, x_2 \geq 0 \end{cases} \tag{7}$$

- $c = [1\ 3]$, $b = [4\ 1\ 3]^T$

$$A = \begin{bmatrix} 1 & 3 \\ 4 & -1 \\ 0 & 1 \end{bmatrix}.$$

- Its dual problem is

$$\text{Max. } 4y_1 + y_2 + 3y_3$$
$$\text{s.t.} \begin{cases} y_1 + 4y_2 \leq 1 \\ 3y_1 - y_2 + y_3 \leq 3 \\ y_1, y_2 \geq 0 \end{cases} \tag{8}$$

- $c = [4\ 1\ 3]$, $b = [1\ 3]^T$

$$A^T = \begin{bmatrix} 1 & 4 & 0 \\ 3 & -1 & 1 \end{bmatrix}.$$

Solve LP by its Dual (3)

$$
\begin{aligned}
&\text{Max.}\;\; 4y_1 + y_2 + 3y_3 \\
&\text{s.t.} \begin{cases} y_1 + 4y_2 \leq 1 \\ 3y_1 - y_2 + y_3 \leq 3 \\ y_1, y_2 \geq 0 \end{cases}
\end{aligned}
\tag{8}
$$

	z	y_1	y_2	y_3	s_1	s_2	f
	1	-4	-1	-3	0	0	0
s_1	0	1	4	0	1	0	1
s_2	0	3	-1	1	0	1	3

Solve LP by its Dual (4)

Round-1

	z	y_1	y_2	y_3	s_1	s_2	f
	1	-4	-1	-3	0	0	0
s_1	0	1	4	0	1	0	1
s_2	0	3	-1	1	0	1	3

$$\Downarrow$$

	z	y_1	y_2	y_3	s_1	s_2	f
	1	0	15	-3	4	0	4
y_1	0	1	4	0	1	0	1
s_2	0	0	-13	1	-3	1	0

Solve LP by its Dual (5)

Round-2

	z	y_1	y_2	y_3	s_1	s_2	f
	1	0	15	-3	4	0	4
y_1	0	1	4	0	1	0	1
s_2	0	0	-13	1	-3	1	0

$$\Downarrow$$

	z	y_1	y_2	y_3	s_1	s_2	f
	1	0	-24	0	-5	3	4
y_1	0	1	4	0	1	0	1
y_3	0	1	4	0	1	0	1

Solve LP by its Dual (6)

Round-3

	z	y_1	y_2	y_3	s_1	s_2	f
	1	0	-24	0	-5	3	4
y_1	0	1	4	0	1	0	1
y_3	0	1	4	0	1	0	1

$$\Downarrow$$

	z	y_1	y_2	y_3	s_1	s_2	f
	1	6	0	0	1	3	10
y_2	0	1	4	0	1	0	1
y_3	0	3.25	0	1	0.25	1	3.25

Solve LP by its Dual (7)

	z	y_1	y_2	y_3	s_1	s_2	f
	1	6	0	0	1	3	10
y_2	0	1	4	0	1	0	1
y_3	0	3.25	0	1	0.25	1	3.25

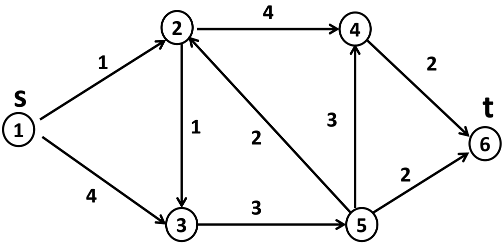
$$
\begin{aligned}
&\text{Min. } x_1 + 3x_2 \\
&\text{s.t.} \begin{cases} x_1 + 3x_2 \geq 4 \\ 4x_1 - x_2 \geq 1 \\ x_2 \geq 3 \\ x_1, x_2 \geq 0 \end{cases}
\end{aligned}
\tag{9}
$$

- The maximum is 10, when $y_1 = 0$, $y_2 = 0.25$ and $y_3 = 3.25$
- According to "**Strong Duality Theorem**", $x_1 + 3x_2 = 10$

Outline

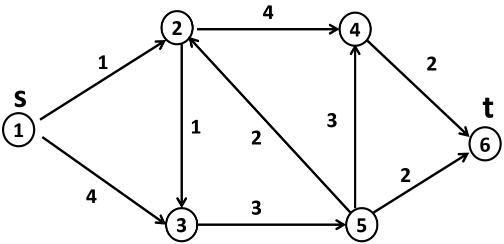
Network Max-flow Problem: the motivation

- Given an oil pipeline network as follows
- What is the maximum sent, per hour, from source to a sink



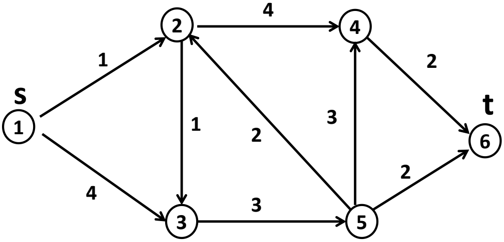
- Given a directed, capacitated network $G = (N, A)$ with arc capacities $u_{ij} \geq 0$, $\forall (i,j) \in A$, determine the maximum possible amount of flow from a designated source node **s** to a sink node **t** while obeying all arc capacities

Network Max-flow Problem: analysis (1)



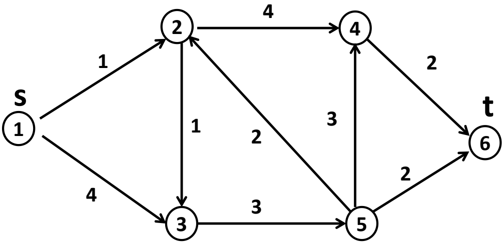
- Given a directed, capacitated network $G = (N, A)$ with arc capacities $u_{ij} \geq 0$, $\forall(i, j) \in A$, determine the maximum possible amount of flow from a designated source node **s** to a sink node **t** while obeying all arc capcities
- What is the objective?
- What are the constraints?

Network Max-flow Problem: analysis (2)



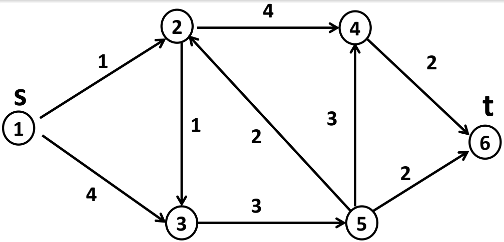
- Let's look at the constraints first!
- Given the maximum flow along arc $\langle i, j \rangle$ is u_{ij}
- Given the planned flow along arc $\langle i, j \rangle$ is x_{ij}
- We have $0 \leq x_{ij} \leq u_{ij}$

Network Max-flow Problem: analysis (3)



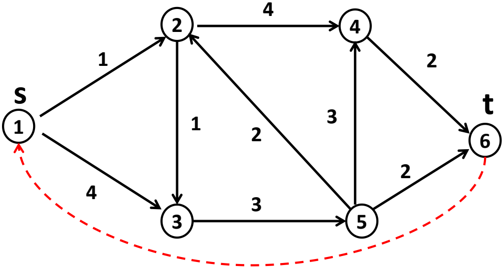
- Let's look at the constraints first!
- For **node 2**, we have $x_{12} + x_{52} = x_{24} + x_{23}$
- For **node 3**, we have $x_{13} + x_{23} = x_{35}$
- For **node 4**, we have $x_{24} + x_{54} = x_{46}$
- For **node 5**, we have $x_{35} = x_{52} + x_{54} + x_{56}$

Network Max-flow Problem: analysis (4)



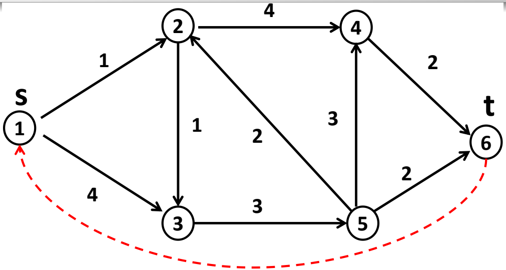
- Let's look at the objective
- We should maximize $x_{12} + x_{13}$ or maximize $x_{46} + x_{56}$
- Under the following constraints
    1. $x_{12} + x_{52} = x_{24} + x_{23}$
    2. $x_{13} + x_{23} = x_{35}$
    3. $x_{24} + x_{54} = x_{46}$
    4. $x_{35} = x_{52} + x_{54} + x_{56}$
    5. $0 \leq x_{ij} \leq u_{ij}$

Network Max-flow Problem: analysis (5)



- Alternatively, we can maximize x_{61}
- Under the following constraints
  1. $x_{12} + x_{52} = x_{24} + x_{23}$
  2. $x_{13} + x_{23} = x_{35}$
  3. $x_{24} + x_{54} = x_{46}$
  4. $x_{35} = x_{52} + x_{54} + x_{56}$
  5. $x_{46} + x_{56} = x_{61}$
  6. $0 \leq x_{ij} \leq u_{ij}$

Network Max-flow Problem: the model



$$\text{Max. } x_{61}$$

$$\text{s.t.} \begin{cases} x_{12} + x_{52} - x_{24} - x_{23} & = 0 \\ x_{13} + x_{23} - x_{35} & = 0 \\ x_{24} + x_{54} - x_{46} & = 0 \\ x_{35} - x_{52} - x_{54} - x_{56} & = 0 \\ x_{46} + x_{56} - x_{61} & = 0 \\ x_{ij} \leq u_{ij} \\ x_{ij} \geq 0 \end{cases} \tag{10}$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% @author: Wan-Lei Zhao
% @date: 2025-9-29
% @solution to maxflow problem
% x12: 1 x13: 2 x23: 3 x24: 4 x35: 5
% x46: 6 x52: 7 x54: 8 x56: 9 x61: 10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f  = [0 0 0 0 0 0 0 0 0 -1];
Ae = [1 0 -1 -1 0 0 1 0 0 0;
       0 1 1 0 -1 0 0 0 0 0;
       0 0 0 1 0 -1 0 1 0 0;
       0 0 0 0 1 0 -1 -1 -1 0;
       0 0 0 0 0 1 0 0 1 -1];
be = [0 0 0 0 0]';
lb = [0 0 0 0 0 0 0 0 0 0]';
ub = [1 4 1 4 3 2 2 3 2 5]';
[x, fval] = linprog(f, [], [], Ae, be, lb, ub)
```