

NOTES OF OPERATION RESEARCH

---

## 运筹学笔记

---

Jinhua Wu



# Contents

<b>1</b>	<b>Optimisation Theory</b>	<b>1</b>
1.1	Linear Programming . . . . .	1
1.1.1	Simplex Method . . . . .	2
1.1.2	Definition . . . . .	2
1.1.3	Prop 1.1 . . . . .	2
1.1.4	Th 1.2 . . . . .	3
1.1.5	Th 1.3 . . . . .	3
1.1.6	Cor 1.4 . . . . .	4
1.1.7	Simplex Method (单纯形法) . . . . .	4
1.1.8	Procedure of simplex method . . . . .	7
1.1.9	Duality (对偶) in linear programming . . . . .	7
1.1.10	Dual simplex method . . . . .	9
1.2	Non-linear programming . . . . .	15
1.2.1	Duality and the Lagrangian . . . . .	20
1.2.2	KT conditions . . . . .	21
1.2.3	Algorithm Methods . . . . .	23
1.3	Integer programming . . . . .	31
1.3.1	B. Cutting plane method . . . . .	33
1.4	Dynamic Programming . . . . .	36
1.4.1	A. Knapsack Problem . . . . .	36
1.4.2	B. 0-1 knapsack Problem . . . . .	39
1.4.3	C. Resource allocation problem . . . . .	41
<b>2</b>	<b>Game Theory</b>	<b>45</b>
2.1	Introduction . . . . .	45
2.2	Two-player zero-sum and constant-sum games . . . . .	46
2.3	Two-player non-constant-sum games . . . . .	54
2.3.1	Cooperative Games . . . . .	58
2.4	Games with $t$ players . . . . .	60

<b>3</b>	<b>Graph Theory and Networks</b>	<b>67</b>
3.1	Terminology . . . . .	67
3.2	Shortest path problem . . . . .	70
3.2.1	Problem 3.1 Shortest path problem . . . . .	70
3.3	Maximum Flow Problem . . . . .	76
3.3.1	Problem 3.5 . . . . .	77
3.4	Minimum Spanning Tree Problem . . . . .	82
3.4.1	Problem 3.9 (Minimum Spanning Tree Problem) . . . . .	83
<b>4</b>	<b>Queueing Theory</b>	<b>87</b>
4.1	Queueing system . . . . .	87
4.2	The exponential distribution . . . . .	91
4.3	Birth-Death Process and M/M/s Systems . . . . .	94

# Chapter 1

## Optimisation Theory

**Assessment:** 70% examination, 30% homework

Six assignments:

7% the first three assignments and 3% the last three assignments.

### 1.1 Linear Programming

LP in standard form:

$$\begin{aligned}\max f &= \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad 1 \leq i \leq m \\ x_j &\geq 0, \quad 1 \leq j \leq n\end{aligned}\tag{1.1}$$

In matrix form:

$$\begin{aligned}\max f &= C^T x \\ Ax &= b \\ x &\geq 0\end{aligned}\tag{1.2}$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$  are given,  $x \in \mathbb{R}^n$  is the **optimisation variable**.  $m < n$ ,  $A$  has the full rank. That is  $\text{rank}(A) = m$ .

- $f = C^T x$  is the objective function.
- $R = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}$  is the feasible set. If  $x \in R$ , then  $x$  is feasible. Thus  $R$  is the intersection of a **plane**  $Ax = b$  with the **positive quadrant**  $x \geq 0$ .

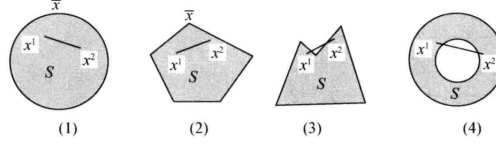


Figure 1.1: In this figure, the panel (1)-(2) are convex. Panel (3)-(4) are not convex.

### 1.1.1 Simplex Method

Aim: we want an effective method to solve the LP problem.

Def: The LP(1.1) is non-degenerate if  $b$  is not a linear combination of  $< m$  columns of  $A$ . In other words, we can't have  $Ax = b$  with  $x$  having  $< m$  non-zeros entries.

*All our problems will be non-degenerate.*

**Def** Give the LP(1.1), a basic solution  $x \in \mathbb{R}^n$  is as follows. Let  $B \in \mathbb{R}^{m \times m}$  be a sub-matrix of  $A$  with full rank  $m$ . Suppose  $B$  occupies columns  $j_1 < j_2 < \dots < j_m$ . Let  $y = B^{-1}b$ . Then

$$x_j = \begin{cases} 0 & \text{if } j \notin \{j_1, j_2, \dots, j_m\} \\ y_l & \text{if } j = j_l \end{cases}$$

Note that  $Ax = by = b$ . If also  $x \geq 0$ , then  $x \in \mathbb{R}$ , and  $x$  is a basic feasible solution.

### 1.1.2 Definition

- $S \subset \mathbb{R}^n$  is convex if  $\forall x, y \in S$  is an extreme point if  $\exists x, y \in S$ ,  $x \neq y$  and  $\lambda \in (0, 1)$ , s.t.  $z = (1 - \lambda)x + \lambda y \in S$ . Note that  $S = \emptyset$  and  $S = \{x\}$  are convex.
- If  $S \subset \mathbb{R}^n$  is convex, then  $z \in S$  is an extreme point if  $\nexists x, y \in S$ ,  $x \neq y$  and  $\lambda \in (0, 1)$ , s.t.  $z = (1 - \lambda)x + \lambda y$ .

### 1.1.3 Prop 1.1

**Proposition 1.1.** In the LP (1.1),  $R$  is convex.

*Proof:* Let  $x, y \in R$  and  $\lambda \in [0, 1]$ . Then

$$A((1 - \lambda)x + \lambda y) = (1 - \lambda)Ax + \lambda Ay = b$$

and

$$(1 - \lambda)x + \lambda y \geq 0$$

since  $x, y \geq 0$  and  $1 - \lambda, \lambda \geq 0$ . Thus,  $(1 - \lambda)x + \lambda y \in R$ .

By Prop 1.1,  $R$  is convex. Thus in 1.2, 1.3 below, we do have the notion of an extreme point of  $R$ .

#### 1.1.4 Th 1.2

**Theorem 1.1.** Suppose the LP (1.1) has an optimal solution. Then  $R$  has an extreme point which is an optimal solution.

*Proof.* Suppose  $x^* \in R$  is optimal, but not extreme. Then we have  $x^* = (1 - \lambda)y + \lambda z$ , for some  $y, z \in R$ ,  $y \neq z$  and  $\lambda \in (0, 1)$ . Now  $c^T x^* = (1 - \lambda)c^T y + \lambda c^T z \leq (1 - \lambda)c^T x^* + \lambda c^T x^* = c^T x^*$ . So  $c^T x^* = c^T y = c^T z$ . Thus,  $c^T((1 - \lambda')y + \lambda'z) = c^T x^*$ ,  $\forall \lambda' \in R$ . That is, all feasible points on the line through  $x^*, y, z$  are optimal.

Let  $u = z - y$ . Then  $A(x^* + \mu u) = Ax^* + \mu Az - \mu Ay = b$ . Since  $u \neq 0$ , assume  $J = \{1 \leq j \leq n : u_j < 0\}$  satisfy  $J \neq \emptyset$ . (Otherwise consider  $u = y - z$ .) Then, we may choose  $\mu' > 0$  maximum such that  $x^* + \mu' u \geq 0$ , that  $(x^* + \mu' u)_j = 0$  for some  $j \in J$ . Now, for  $1 \leq j \leq n$ ,  $x_i^* = 0 \implies y_i = z_i = 0$ . (since  $x_i^* = (1 - \lambda)y_i + \lambda z_i$  and  $y_i z_i \geq 0$ ) That means that  $\implies u_i = 0 \implies (x^* + \mu' u)_i = 0$ . Thus  $i \notin J$  for such  $i$ . We see that  $x^* + \mu' u \in R$  has more zero coordinates than  $x^*$ , and  $x^* + \mu' u$  is also optimal.

Repeat this argument, we must stop since we can't have more than  $n$  zeros. Upon termination, we have an extreme point of  $R$ , which is optimal.  $\square$

#### 1.1.5 Th 1.3

**Theorem 1.2.** Suppose the LP (1.1) is non-degenerate. Let  $x \in R$ , then  $x$  is extreme  $\iff x$  is a basic feasible solution.

*Proof.*  $\implies$  Suppose  $x \in R$  is extreme, but not a basic feasible solution. Let  $J = \{1 \leq j \leq n : x_j > 0\}$ . Suppose first that  $|J| \leq m$ . Then either  $|J| < m$  or  $|J| = m$ . And the columns of  $A$  indexed by  $J$  are linearly dependent. In either case,  $b$  is a linear combination of less than  $m$  columns of  $A$ , which contradicts non-degeneracy.

Now, let  $|J| > m$ . Then the columns of  $A$  indexed by  $J$  are linearly dependent. If  $A'$  is the submatrix of  $A$  with these columns, then  $\exists v' \in \mathbb{R}^{|J|} \setminus \{0\}$  such that  $A'v' = 0$ . Extend  $v'$  to  $v \in \mathbb{R}^n \setminus \{0\}$  by adding zeros to the remaining coordinators (not indexed by  $J$ ). Then  $Av = 0$ , and  $x_i = 0 \implies v_i = 0$ . Now,  $x \pm \varepsilon v \in R$  if  $\varepsilon > 0$  is small:  $A(x \pm \varepsilon v) = Ax = b$ , and  $x \pm \varepsilon v \geq 0$  for  $\varepsilon > 0$  small (since  $x_i = 0 \implies v_i = 0$ ). Thus  $x = \frac{1}{2}(x + \varepsilon v) + \frac{1}{2}(x - \varepsilon v)$ , contradicting  $x$  extreme.  $\square$

$\Leftarrow$  Suppose  $x \in R$  is a basic feasible solution, but not extreme. Then  $x = (1 - \lambda)y + \lambda z$  for some  $y, z \in R$ ,  $y \neq z$  and  $\lambda \in (0, 1)$ . Let  $B \in \mathbb{R}^{m \times m}$  be the submatrix of  $A$  corresponding to  $x$ , and  $J \subset \{1, 2, \dots, n\}$  be the index set of the columns of  $B$ . We have  $|J| = m$  and  $\text{rank}(B) = m$ . For  $i \notin J$ , we have  $x_i = 0 \implies y_i = z_i = 0$  (as in **Th 1.2**). Let  $y', z' \in \mathbb{R}^m$  be obtained from  $y, z$  by deleting the zeros

corresponding to  $\{1, 2, \dots, n\} \setminus J$ . Then  $b = Ay = By'$ ,  $b = Az = Bz' \Rightarrow B(y' - z') = 0$ . But  $y' - z' \neq 0$ , so  $\text{rank}(B) < m$ , contradiction.

### 1.1.6 Cor 1.4

Now 1.1.4 and 1.1.5  $\Rightarrow$

**Corollary 1.1.1.** *Suppose the LP(1.1) has an optimal solution and is non-degenerate. Then  $\exists$  basic feasible solution which is optimal.*

### 1.1.7 Simplex Method (单纯形法)

Now we present the **simplex method** for solving the LP(1.1). By 1.1.6, to find the maximum of  $f$ , it suffices to check the value of  $f$  at basic feasible solutions. There are  $\leq \binom{n}{m}$  such points, since there are  $\leq \binom{n}{m}$  choices for the submatrix  $B$  of  $A$ . But the idea is, we move from one basic feasible solution to another, and strictly increase  $f$  each time. When this procedure stops, we either have an optimal solution, or conclude that there are none.

In (1.1), we treat  $f$  as another variable. We have the system:

$$\begin{cases} Ax = b \\ -c^T x + f = 0 \\ x \geq 0 \end{cases}$$

The system is the initial system of the LP 1.1. We describe the simplex method, using Example 1.

**Example 1** Again consider Example 1. which has the standard from 1.2. The initial system is:

$$\begin{cases} x_1 & +x_2 & +s_1 & & & = 6 \\ 2x_1 & +4x_2 & & +s_2 & & = 16 \\ & x_2 & & & +s_3 & = 3 \\ -3x_1 & -4x_2 & & & & +f = 0 \\ x_1, & x_2, & s_1, & s_2, & s_3 & \geq 0 \end{cases}$$

We form the simplex table as follows:

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$f$	
1	1	1	0	0	0	6
2	4	0	1	0	0	16
0	1	0	0	1	0	3
-3	-4	0	0	0	1	0

Note that  $(x_1, x_2, s_1, s_2, s_3) = (0, 0, 6, 16, 3)$  is a basic feasible solution, and  $f = 0$ . We require that the entries on the right are  $\geq 0$ . We choose an entry called the **pivot** as follows.



① Choose the column containing the most negative entry in the bottom row, containing  $-4$ . We choose this since the objective function is  $f = 3x_1 + 4x_2$ , so increasing along  $x_2$  is the most beneficial.

② Next, for each positive entry in the chosen column, (except for the bottom one), divide the number on the right by the entry. Choose the row with the minimum answer, which is the third row, with  $3/1 = 3$ . We choose this row so that all three equality constraints of 1.2 can be satisfied when  $x_2 \leq 3$ .

The **pivot** is the entry in the chosen column and row.

③ We now reduce by making the pivot 1 (here, it already is) and all other entries in the second column 0. We do not switch rows.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$f$	
1	0	1	0	-1	0	3
2(pivot)	0	0	1	-4	0	4
0	1	0	0	1	0	3
<b>-3</b>	0	0	0	-4	1	12

We have now moved to the basic feasible solution  $(x_1, x_2, s_1, s_2, s_3) = (0, 3, 3, 4, 0)$  and  $f = 12$ . The value of  $f$  has increased.

If we still have negative entry in the entry in the bottom row, repeated the procedure. Continue until this ends.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$f$	
0	0	1	-1/2	1	0	1
1	0	0	1/2	-2(pivot)	0	2
0	1	0	0	1	0	3
0	0	0	3/2	<b>-2</b>	1	18

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$f$	
0	0	1	-1/2	1	0	1
1	0	2	1/2	0	0	4
0	1	-1	0	0	0	2
0	0	0	2	1/2	1	20

We end at this table since there are no more negative entries in the bottom row, which gives:

$$f = 20 - 2s_2 - \frac{1}{2}s_3 \leq 20 \quad (1.3)$$

We can read off the final, optimal solution as  $(x_1, x_2, s_1, s_2, s_3) = (4, 2, 0, 0, 1)$  and  $f = 20$ . This agrees with the answer in Example 1.

What happens if a LP has no optimal solution?

**Example 1.1.**

$$\begin{aligned}
\min \quad & f = -5x_1 + x_2 \\
& x_1 - x_2 \leq 2 \\
& -3x_1 + x_2 \leq 0 \\
& x_1, x_2 \geq 0
\end{aligned}$$

*Solution.* We change the objective function to maximize  $f' = 5x_1 - x_2$ .

Initial system:

$$\begin{cases}
x_1 - x_2 + s_1 &= 2 \\
-3x_1 + x_2 + s_2 &= 0 \\
-5x_1 + x_2 + f' &= 0 \\
x_1, x_2, s_1, s_2 &\geq 0
\end{cases}$$

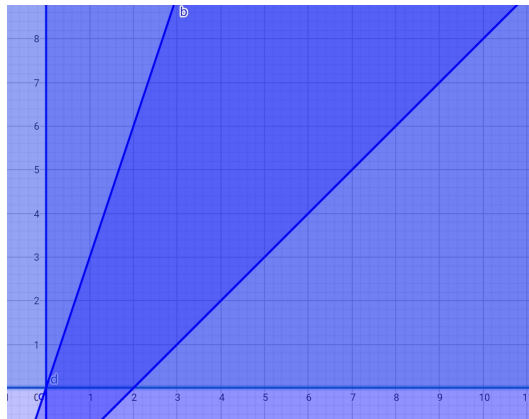
Simplex table:

$x_1$	$x_2$	$s_1$	$s_2$	$f$	
1(pivot)	-1	1	0	0	2
-3	1	0	1	0	0
<b>-5</b>	1	0	0	1	0

$x_1$	$x_2$	$s_1$	$s_2$	$f$	
1	-1	1	0	0	2
0	-2	3	1	0	6
0	-4	5	0	1	10

Since no entry in the second column is positive, the LP does not have a solution: there is no finite maximum for  $f'$ , hence no finite minimum for  $f$ . To see this, set  $s_1 = 0$ , since  $s_1$  is a non-basic variable. Set  $x_2 = M \geq 0$ . Then  $(x_1, x_2, s_1, s_2) = (2 + M, M, 0, 6 + 2M)$  is feasible, and  $f' = 10 + 4M \rightarrow \infty$  as  $M \rightarrow \infty$ .

Picture of LP problem:





### 1.1.8 Procedure of simplex method

**Procedure 1.1.1.** Give a LP:

- (1) Convert the LP to standard form, introducing slack variables and changing min to max, if necessary. Write down the initial system.
- (2) Write down the simplex table. The entries on the right are  $\geq 0$ , and some columns form on identity matrix, which gives a basic feasible solution.
- (3) If there is a negative number in the bottom row, choose the column with the most negative number. If there is a tie, choose any such column. If there is no negative number, the optimal solution of the LP is found. Read off the values of the optimal solution and the optimal value.
- (4) For every positive entry in the chosen column from Step 3(except the bottom one), divide the number on the right by the entry. Choose the row with the minimum answer. If there is a tie, choose any such row. If there is no positive entry in the column, then the LP has no optimal solution.
- (5) The entry in the column and row from Step 3 and 4 is the pivot. Perform row operations to make the pivot 1, and other entries in the column 0. Do not exchange rows.
- (6) Repeat Step 3 for the new simplex table. Continue until termination.

**Remark 1.1.** We note that the simplex method does indeed terminate. There are finitely many basic feasible solutions,  $\leq \binom{n}{m}$ . Enough to show that no basic feasible solution is visited more than once. For this, enough to show that  $f$  strictly increases in every move. At an iterate, the current value of  $f$ , say  $P$ , is in the bottom right of the simplex table. If the pivot is in row  $i$  and  $P_i \geq 0$  is the entry in the right of row  $i$ , then the next value for  $f$  is  $P' = P + kP_i$  for some  $k > 0$ . If  $P_i = 0$ , then we have a basic feasible solution with  $< m$  non-zero entries, which contracts non-degeneracy. Hence,  $P_i > 0$  and  $P' = P + kP_i > P$ .

### 1.1.9 Duality (对偶) in linear programming

For  $x, y \in \mathbb{R}^n$ , write  $x \geq y$  to mean  $x - y \geq 0$ . We write  $x \geq y$  to mean  $y \geq x$ .

**Remark 1.2.** Give a LP, we want to associate it with another LP, called the dual.

Consider the LP:

$$\begin{aligned} \max \quad & c^T \mathbf{x} \\ & A\mathbf{x} \leq b \\ & \mathbf{x} \geq 0 \end{aligned} \tag{1.4}$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$  are given, and  $\mathbf{x} \in \mathbb{R}^n$  is variable. Let  $p^* \in \mathbb{R}$  be the optimal value

**Remark 1.3.**  $p^* = -\infty$  if (1.4) is not feasible, otherwise  $p^*$  is finite or  $p^* = +\infty$ .

We introduce a variable  $\mathbf{y} \in \mathbb{R}^m$ ,  $\mathbf{y} \geq 0$ . Then,  $\forall \mathbf{x} \in \mathbb{R}^n$  feasible for (1.4).

$$\mathbf{y}^T A\mathbf{x} \leq \mathbf{y}^T b \quad \text{or} \quad (A^T \mathbf{y})^T \mathbf{x} \leq b^T \mathbf{y} \tag{1.5}$$

If  $\mathbf{y}$  also satisfies  $A^T \mathbf{y} \geq c$ , then

$$c^T \mathbf{x} \leq (A^T \mathbf{y})^T \mathbf{x} \leq b^T \mathbf{y} \tag{1.6}$$

Thus, we want to find  $\mathbf{y}$  such that (1.6) is best possible. We define the **dual** LP of the LP (1.4) to be:

$$\begin{aligned} \min \quad & b^T \mathbf{y} \\ & \begin{cases} A^T \mathbf{y} \geq c \\ \mathbf{y} \geq 0 \end{cases} \end{aligned} \tag{1.7}$$

where  $\mathbf{y} \in \mathbb{R}^m$  is a variable. Let  $d^*$  be the optimal value.

**Remark 1.4.**  $d^* = \infty$  if (1.7) is not feasible, otherwise,  $d^*$  is finite or  $d^* = -\infty$ .

The LP (1.4) is the **primal** LP.

**Proposition 1.2.** The dual LP of the LP (1.7) is the LP (1.4).

*Solution.* By writing (1.7) as:

$$\begin{aligned} \max \quad & -b^T \mathbf{y} \\ & -A^T \mathbf{y} \leq c \\ & \mathbf{y} \geq 0 \end{aligned}$$

the dual is

$$\begin{aligned} \min \quad & -c^T \mathbf{x} \\ & -A\mathbf{x} \geq -b \\ & \mathbf{x} \geq 0 \end{aligned}$$

which is (1.4). □

**Theorem 1.3.** Weak Duality: Given LP (1.4) and its dual (1.7), then we have:

- (a)  $p^* \leq d^*$ ;
- (b) If  $p^* = \infty$ , then (1.7) is not feasible.
- (c) If  $d^* = \infty$ , then (1.4) is not feasible.
- (d) Suppose  $\bar{\mathbf{x}} \in \mathbb{R}^n$ ,  $\bar{\mathbf{y}} \in \mathbb{R}^m$  are feasible for (1.4) and (1.7), and  $c^T \bar{\mathbf{x}} = b^T \bar{\mathbf{y}}$ . Then  $c^T \bar{\mathbf{x}} = p^* = d^* = b^T \bar{\mathbf{y}}$ .

*Solution.* (a) If either (1.4) or (1.7) is not feasible, then  $p^* \leq d^*$  holds. Suppose (1.4) and (1.7) are both feasible. We have  $c^T \mathbf{x} \leq b^T \mathbf{y}$ ,  $\forall$  feasible  $\mathbf{x} \in \mathbb{R}^n$  for (1.4),  $\forall$  feasible  $\mathbf{y} \in \mathbb{R}^m$  for (1.7). Thus  $p^* \leq b^T \mathbf{y}$ ,  $\forall$  feasible  $\mathbf{y} \in \mathbb{R}^m$  for (1.7) and  $p^* \leq d^*$ .

(b) If (1.7) is feasible, then  $p^* \leq b^T \mathbf{y}$ , for all feasible  $\mathbf{y} \in \mathbb{R}^m$  for (1.7), and thus  $p^* < \infty$ .

(c) Similar to (b).

(d) By (a), we have  $c^T \bar{\mathbf{x}} \leq p^* \leq d^* \leq b^T \bar{\mathbf{y}} = c^T \bar{\mathbf{x}}$ , so equality holds throughout. □

**Theorem 1.4.** Strong Duality: Given LP in (1.4) and its dual (1.7). If either LP is feasible and bounded, then so is the other, and  $p^* = d^*$ .

**Remark 1.5.** The LP (1.4) is BOUNDED if  $p^*$  is finite. Similarly for the LP (1.7) with  $d^*$ .

We will prove TH 1.7 later.

### 1.1.10 Dual simplex method

**Example 1.2.** Consider the LP:

$$\begin{aligned} \max \quad & f = -4x_1 - 2x_2 - x_3 \\ & \begin{cases} -x_1 - x_2 + 2x_3 & \leq -3 \\ -4x_1 - 2x_2 + x_3 & \leq -4 \\ x_1 + x_2 - 2x_3 & \leq 2 \\ x_1, x_2, x_3 & \geq 0 \end{cases} \end{aligned} \tag{1.8}$$

Introducing slack variables  $s_1, s_2, s_3 \geq 0$ , we have the form:

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$s_3$	$f$	
-1	-1	2	1	0	0	0	-3
-4	-2	1	0	1	0	0	-4
1	1	-4	0	0	1	0	2
4	2	1	0	0	0	1	0

Table 1.1: table-1

But this says the basic solution is  $(x_1, x_2, x_3, s_1, s_2, s_3) = (0, 0, 0, -3, -4, 2)$ , which is not feasible for (1.8). We can't use the simplex method.

We note that the dual of (1.8) is

$$\begin{aligned} \max \quad & y = -3y_1 - 4y_2 + 2y_3 \\ & \begin{cases} -y_1 - 4y_2 + y_3 \leq -4 \\ -y_1 - 2y_2 + y_3 \leq -2 \\ 2y_1 + y_2 - 4y_3 \leq -1 \\ y_1, y_2, y_3 \geq 0 \end{cases} \end{aligned} \tag{1.9}$$

Converting (1.9) to standard form, we can solve it by the simplex method. Then the **Th 1.7**  $\implies$  the solution to (1.9) gives the solution to (1.8).

However, the dual simplex method can solve both (1.8) and (1.9) simultaneously. We demonstrate the method with the example.

**Procedure 1.1.2. STEP 1** Write out the dual simplex tables.

	$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$s_3$	
$s_1$	-1	-1	2	1	0	0	-3
$s_2$	-4	-2	1	0	1	0	-4
$s_3$	1	1	-4	0	0	1	2
$-f$	-4	-2	-1	0	0	0	0

$y = 0$  is dual feasible simplex since all numbers are  $\leq 0$ .

**STEP 2** We aim for an operation about a pivot so that the primal LP “goes toward being feasible”, while the dual LP remains feasible. Choose any row with a negative entry on the right, say row 1. This means that we will increase  $y$ , which decreases the value of the dual objective. Next, for every column with a negative entry in the chosen row (row 1), divide the entry in the bottom row by the entry (row 1). We obtain  $\frac{-4}{-1} = 4$  and  $\frac{-2}{-1} = 2$ . Choose the column smallest answer, which is positive. This says that  $y_1$  can increase by at most 2, so that the constraints of (1.9) are satisfied.

**STEP 3** Row reduce as before. We obtain:

	$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$s_3$	
$x_2$ replace with variable in pivot column	1	1	-2	-1	0	0	3
$s_2$	-2	0	-3	-2	1	0	2
$s_3$	0	0	-2	1	0	1	-1
$-f$	-2	0	-5	-2	0	0	6

Note that this new table is still dual feasible, simplex choice of the pivot implies all entries in the bottom row remain  $\leq 0$ . The primal has become “towards feasible”, since now there is only one negative number on the right.

**STEP 4** Repeat until procedure ends, when all numbers on the right are  $\geq 0$

	$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$s_3$	
$x_2$	1	1	0	-2	0	-1	4
$\rightarrow s_2$	-2	0	0	$-\frac{7}{2}$	1	$-\frac{3}{2}$	$\frac{7}{2}$
$x_3$	0	0	1	$-\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$
$-f$	-2	0	0	$-\frac{9}{2}$	0	$-\frac{5}{2}$	$\frac{17}{2}$

Final optimal solutions:  $(x_1, x_2, x_3, s_1, s_2, s_3) = (0, 4, \frac{1}{2}, 0, \frac{7}{2}, 0)$ , and  $f = -\frac{17}{2}$ ,  $(y_1, y_2, y_3) = (\frac{9}{2}, 0, \frac{5}{2})$ .

**Remark 1.6.** In **STEP 2**, we could have chosen row 2.

If there is a tie for the ratios in **STEP 2**, we could choose any suitable column.

As before, if the dual simplex method does not terminate, we conclude that there is no optimal solution.

**Example 1.3.** Use the dual simplex method to solve the LP:

$$\begin{aligned} \min \quad & f = 2x_1 + 3x_2 + x_3 \\ \left\{ \begin{array}{l} x_1 + x_2 - 2x_3 \geq 3 \\ x_1 + 2x_2 - x_3 \leq 2 \\ x_1, x_2, x_3 \geq 0 \end{array} \right. \end{aligned} \quad (1.10)$$

We change the objective to  $\max f' = -2x_1 - 3x_2 - x_3$ , and add slack variables  $s_1, s_2 \geq 0$ . We obtain:

	$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	
$s_1$	-1( <b>PIVOT</b> )	-1	2	1	0	-3
$s_2$	-1	-2	1	0	1	-2
$-f'$	-2	-3	-1	0	0	0

Thus we have:

	$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	
$s_1$	1	1	-2	-1	0	3
$s_2$ ( <b>NO NEGATIVE NUMBER</b> )	0	-1	-1	-1	1	-1
$-f'$	0	-1	-5	-2	0	6

This means the dual LP is unbounded. By Th 1.6, the primal LP(1.10) is not feasible.

**Example 1.4.**

$$\begin{aligned} \min \quad & f = -3x_1 + x_2 - 2x_3 \\ \left\{ \begin{array}{l} x_1 + 3x_2 + x_3 \leq 5 \\ -2x_1 + x_2 - x_3 \leq -2 \\ 4x_1 - x_2 - 2x_3 = 5 \\ x_1, x_2, x_3 \geq 0 \end{array} \right. \end{aligned} \quad (1.11)$$

By writing the objective as  $\max f' = -3x_1 + x_2 - 2x_3$ , and adding slack variables  $s_1, s_2 \geq 0$ , we



find

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$f'$	
1	3	1	1	0	0	5
-2	1	-1	0	1	0	-2
4	-1	2	0	0	0	5
-3	1	-2	0	0	-1	0

We can't use the simplex method or dual simplex method directly, since both of sets of coefficients  $\{5, -2, 5\}$  and  $\{-3, 1, -2\}$  have negative numbers.

We see that the solution  $(x_1, x_2, x_3, s_1, s_2) = (0, 0, 0, 5, -2)$  is not feasible. We apply the **TWO-PHASE METHOD** where **PHASE 1** tries to find a basic solution.

**PHASE 1** We form the problem:

$$\begin{aligned} \max \quad & h = -t_1 - t_2 \\ \left\{ \begin{array}{lcl} x_1 + 3x_2 + x_3 + s_1 & = & 5 \\ 2x_1 - x_2 + x_3 - s_2 + t_1 & = & 2 \\ 4x_1 - x_2 - 2x_3 + t_2 & = & 5 \end{array} \right. \end{aligned} \quad (1.12)$$

where  $s_1 \geq 0$  is a slack variable,  $s_2 \geq 0$  is a surplus variable and  $t_1, t_2 \geq 0$  are artificial variables. If the optimal solution of (1.12) is  $h = 0$ , then  $\exists x_1, x_2, x_3, s_1, s_2 \geq 0$ , s.t. the constraints of (1.12) hold, with  $t_1 = t_2 = 0$ . This will be a basic feasible solution for (1.11). We form an extended version of simplex table.

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$t_1$	$t_2$	$f'$	$h$	
1	3	1	1	0	0	0	0	0	5
2	-1	1	0	-1	1	0	0	0	2
4	-1	-2	0	0	0	1	0	0	5
-3	1	-2	0	0	0	0	-1	0	0
0	0	0	0	0	1	1	0	1	0

the bottom entries in the  $t_1, t_2$  columns 0. Then we perform the simplex method procedure.

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$t_1$	$t_2$	$f'$	$h$	
1	3	1	1	0	0	0	0	0	5
2(pivot)	-1	1	0	-1	1	0	0	0	2
4	-1	-2	0	0	0	1	0	0	5
-3	1	-2	0	0	0	0	-1	0	0
-6	2	1	0	1	0	0	0	1	7

Thus we have:

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$t_1$	$t_2$	$f'$	$h$	
0	$\frac{7}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	4
1	$-\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	1
0	1	-4	0	2	-2	1	0	0	1
0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$-\frac{3}{2}$	$\frac{3}{2}$	0	1	0	3
0	-1	4	0	-2	3	0	0	1	-1

Feb, Mar 12

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$t_1$	$t_2$	$f'$	$h$	
0	$\frac{13}{4}$	$\frac{3}{2}$	1	0	0	$-\frac{1}{4}$	0	0	$\frac{15}{4}$
1	$-\frac{1}{4}$	$-\frac{1}{2}$	0	0	0	$\frac{1}{4}$	0	0	$\frac{5}{4}$
0	$\frac{1}{2}$	-2	0	1	-1	$\frac{1}{2}$	0	0	$\frac{1}{2}$
0	$\frac{1}{4}$	$-\frac{7}{2}$	0	0	0	$\frac{3}{4}$	1	0	$\frac{15}{4}$
0	0	0	0	0	1	1	0	1	0

This now says that we have the basic feasible solution  $(x_1, x_2, x_3, s_1, s_2) = (\frac{5}{4}, 0, 0, \frac{15}{4}, \frac{1}{2})$ , and  $t_1 = t_2 = h = 0$ . If we could not reach this point, then the optimal solution of (1.12) would be negative and (1.11) would not be feasible.

**PHASE 2** We may not neglect  $t_1, t_2, h$  and use the simplex method.

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$f'$	
0	$\frac{13}{4}$	$\frac{3}{2}$ (Pivot)	1	0	0	$\frac{15}{4}$
1	$-\frac{1}{4}$	$-\frac{1}{2}$	0	0	0	$\frac{5}{4}$
0	$\frac{1}{2}$	-2	0	1	0	$\frac{1}{2}$
0	$\frac{1}{4}$	$-\frac{7}{2}$	0	0	1	$\frac{15}{4}$

Thus we have:

$x_1$	$x_2$	$x_3$	$s_1$	$s_2$	$f'$	
0	$\frac{13}{6}$	1	$\frac{2}{3}$	0	0	$\frac{5}{2}$
1	$\frac{5}{6}$	0	$\frac{1}{3}$	0	0	$\frac{5}{2}$
0	$\frac{29}{6}$	0	$\frac{4}{3}$	1	0	$\frac{11}{2}$
0	$\frac{47}{6}$	0	$\frac{7}{6}$	0	1	$\frac{25}{2}$

Final solution for (1.11):  $\min f = -\frac{25}{2}$ , when  $(x_1, x_2, x_3) = (\frac{5}{2}, 0, \frac{5}{2})$ . And  $(s_1, s_2) = (0, \frac{11}{2})$ .

*Solution.* For Th 1.7

By Prop 1.5 and Th 1.6(d), it suffices to assume LP(1.4) is feasible and bounded, and prove exists feasible  $\bar{x} \in \mathbb{R}^n$  and  $\bar{y} \in \mathbb{R}^m$  for (1.4) and (1.7), s.t.  $c^T \bar{x} = b^T \bar{y}$ . We apply either the simplex or two-Phase method to (1.4) by introducing  $x \in \mathbb{R}^m$  which contains slack and surplus method. The simplex or two-phase method moves from one point of the set  $S = \{ \begin{pmatrix} x \\ s \end{pmatrix} \in \mathbb{R}^{n+m} : Ax + s = b \}$  to another, which varying the objective function value  $c^T x$ . When the algorithm ends, an optimal solution for (1.4) is found, and the objective function reads  $p^* - \bar{z}^T \bar{x} - \bar{y}^T \bar{s}$  for some  $\bar{x}, \bar{z} \in \mathbb{R}^n$ ,  $\bar{s}, \bar{y} \in \mathbb{R}^m$  with  $A\bar{x} + \bar{s} = b$  and  $\bar{x}, \bar{z}, \bar{s}, \bar{y} \geq 0$ . Thus  $\forall \begin{pmatrix} x \\ s \end{pmatrix} \in S$ , we have  $p^* - \bar{z}^T x - \bar{y}^T s = c^T x$ , so  $\forall x \in \mathbb{R}^n$ , we have:

$$p^* - \bar{z}^T x - \bar{y}^T (b - Ax) = c^T x.$$

Equating the constant terms gives  $p^* = b^T \bar{y}$ . Equating the  $x$  terms gives  $A^T \bar{y} = c + \bar{z} \geq c$ . So  $\bar{y}$  is the feasible for (1.7). Finally, when the simplex or two-Phase method terminates, all variables with a non-zero coefficient in  $\bar{z}^T \bar{x} + \bar{y}^T \bar{s}$  is a non-basic variable, and hence equals zero in the solution  $\begin{pmatrix} \bar{x} \\ \bar{s} \end{pmatrix}$ . Thus,

$$p^* = p^* - \bar{z}^T \bar{x} - \bar{y}^T \bar{s} = c^T \bar{s}$$

and  $c^T \bar{x} = b^T \bar{y}$ , as required. □

**Remark 1.7.** We developed the results and methods as follows. There were no circular chain of implications:

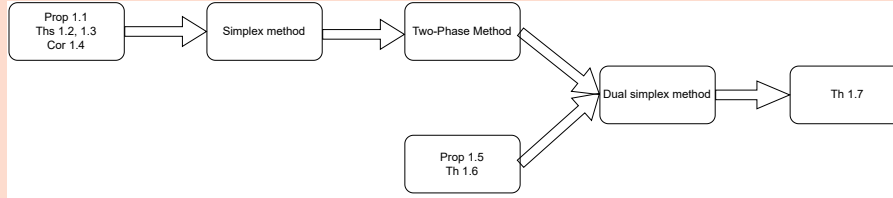


Figure 1.2: The chain of implications

Even though the two-phase method seems stronger than the dual simplex method, we actually derived the latter from the former!

## 1.2 Non-linear programming

**Definition 1.1.** convex: Let  $X \in \mathbb{R}^n$  be convex. A function  $f : X \rightarrow \mathbb{R}$  is convex if  $\forall x, y \in X$  and  $\lambda \in [0, 1]$

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y)$$

$f$  is strictly convex if  $\forall x, y \in X$  with  $x \neq y$ , and  $\lambda \in (0, 1)$

$$f((1 - \lambda)x + \lambda y) < (1 - \lambda)f(x) + \lambda f(y)$$

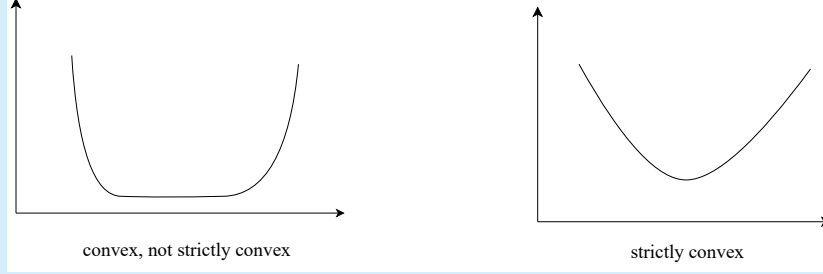


Figure 1.3: Convex function

$f$  is concave/strictly concave if  $-f$  is convex/strictly convex.

**Definition 1.2.** Differentiable: Let  $X \in \mathbb{R}^n$  be an open set. A function  $f : X \rightarrow \mathbb{R}$  is  $k$  times differentiable if all partial derivatives:

$$\frac{\partial^l f}{\partial x_1^{l_1} \partial x_2^{l_2} \cdots \partial x_n^{l_n}} \quad (1.13)$$

exist, where  $0 \leq l_i, l \leq k$  and  $l_1 + l_2 + \cdots + l_n = l$ . Recall that  $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$ ,  $\forall 1 \leq i, j \leq n$ ,  $f$  is  $k$  times continuously differentiable if in addition, all derivatives (1.13) are continuous on  $X$ . If  $k = 1$ , we omit “ $k$  times”. If  $k = 2$ , we write “twice” for “2 times”.

The gradient of  $f$  is

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$$

The Hessian of  $f$  is

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}, \quad \text{which is symmetric.}$$

**Definition 1.3** (symmetric definite matrix). A symmetric matrix  $A \in \mathbb{R}^{n \times n}$  is:

Positive definite (正定) Written  $A > 0$ , if  $x^T A x > 0$ ,  $\forall x \in \mathbb{R}^n \setminus \{0\}$ .

Positive semi-definite Written  $A \geq 0$ , if  $x^T A x \geq 0$ ,  $\forall x \in \mathbb{R}^n$ .

The following results each characterises convex functions.

**Theorem 1.5.** Let  $f : X \rightarrow \mathbb{R}$ , where  $X \subset \mathbb{R}^n$  is convex. Then  $f$  is convex  $\iff \forall x, y \in X$ ,  $\forall \lambda \in [0, 1]$ , the function:

$$h(\lambda) = f((1 - \lambda)x + \lambda y)$$

is convex on  $\lambda \in [0, 1]$ . That is,  $f$  is convex when restricted to any line segment in  $X$ .

**Theorem 1.6.** Let  $f : X \rightarrow \mathbb{R}$  be differentiable, where  $X \subset \mathbb{R}$  is open. Then  $f$  is convex  $\iff X$  is convex, and

$$f(y) \geq f(x) + \nabla f(x)^T(y - x)$$

“ $f$  lies above the plane of contact at  $x$ ”

**Theorem 1.7.** Let  $f : X \rightarrow \mathbb{R}$  be twice differentiable, where  $X \subset \mathbb{R}$  is open. Then  $f$  is convex  $\iff X$  is convex, and

$$\nabla^2 f(x) \geq 0, \quad \forall x \in X$$

The standard form of a non-linear programming problem is

$$\begin{aligned} \min \quad & f(x) \\ & g_i(x) \leq 0, \quad \forall 1 \leq i \leq m \end{aligned} \tag{1.14}$$

where  $x \in \mathbb{R}^n$  is the optimisation variable, and  $f : X_0 \rightarrow \mathbb{R}$ ,  $g_i : X_i \rightarrow \mathbb{R}$ , where  $X_0, X_1, \dots, X_m \subset \mathbb{R}^n$  are the domains of  $f, g_1, \dots, g_m$ . The function  $f$  is the objective function, and  $g_i(x) \leq 0$  are the (inequality) constraints. The domain and feasible set of problem (1.14) are

$$\begin{aligned} D &= \bigcap_{i=0}^m X_i \\ R &= \{x \in D : g_i(x) \leq 0, \forall 1 \leq i \leq m\} \end{aligned}$$

The optimal value of (1.14) is denoted by  $p^*$ . The optimal set of (1.14) is

$$R^* = \{x \in R : f(x) = p^*\}$$

Any elements of  $R^*$  is an **OPTIMAL SOLUTION**. Note that we may have  $R^* = \emptyset$ , even if  $p^*$  is finite.

**Example 1.5.** If  $f(x) = e^{-x}$ , then  $p^* = 0$ , but  $R^* = \emptyset$ .

Think of a series, its limitation isn't located on its set. Commonly appearing in Real Analysis and some courses. Secondly perhaps the function isn't continuous.

If  $R^* \neq \emptyset$ , then  $p^*$  is **OTTAINED** (by any element of  $R^*$ ).

If  $f$  and all  $g_i$  in (1.14) are convex functions (so all sets  $X_i$  are convex), then (1.14) is a **CONVEX PROGRAMMING (CP) PROBLEM**.

**Theorem 1.8.** Th 1.11 Suppose (1.14) is a CP.

- (a) The sets  $R$  and  $R^*$  are convex.
- (b) If  $F$  is strictly convex, and  $R^* \neq \emptyset$ , then  $\exists$  unique optimal solution.

*Proof.* (a) Let  $x, y \in R$ , and  $\lambda \in [0, 1]$ . Then  $x, y \in D$  and since  $D = \bigcap_{i=1}^m X_i$  is convex<sup>1</sup>,  $z = (1 - \lambda)x + \lambda y \in D$ . Now  $1 \leq i \leq m$ , since  $g_i$  is convex. Then

$$g_i(z) = g_i((1 - \lambda)x + \lambda y) \leq (1 - \lambda)g_i(x) + \lambda g_i(y) \leq 0,$$

so  $z \in R$  and  $R$  is convex.

Now let  $x, y \in R^*$ , and  $\lambda \in [0, 1]$ . Then  $x, y \in R$  and since  $R$  is convex, we have  $z = (1 - \lambda)x + \lambda y \in R$ . Since  $f$  is convex,

$$p^* \leq f(z) = f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y) = p^*$$

So  $f(z) = p^*$  and  $z \in R^*$ . Thus  $R^*$  is also convex.

(b) Suppose that  $\exists$  optimal solutions  $x, y \in R^*$  with  $x \neq y$ . Let  $\lambda \in (0, 1)$ . Then  $(1 - \lambda)x + \lambda y \in R^*$  by (a), and

$$f((1 - \lambda)x + \lambda y) < (1 - \lambda)f(x) + \lambda f(y) = p^*$$

a contradiction. □

**Definition 1.4.** In problem (1.14),  $\bar{x} \in R$  is locally optimal if  $\exists \delta > 0$ , s.t.

$$f(x) \leq f(\bar{x}), \quad \forall x \in R \cap B^0(\bar{x}, \delta) \tag{1.15}$$

where  $B^0(\bar{x}, \delta) = \{y \in \mathbb{R}^n : \|y - \bar{x}\| < \delta\}$  is the open ball with radius  $\delta$ , centre  $\bar{x}$ .

---

<sup>1</sup>By HW1 in the homework

**Remark 1.8.** Recall that  $\|z\| = \|z\|_2 = \left( \sum_{i=1}^n |z_i|^2 \right)^{1/2}$  is the Euclidean norm of  $z \in \mathbb{R}^n$

**Theorem 1.9.** Th 1.12 Suppose (1.14) is a CP. Then  $\bar{x}$  is locally optimal  $\iff \bar{x}$  is an optimal solution.

*Proof.* Clearly it suffices to prove ( $\implies$ ). Let  $\bar{x}$  be locally optimal. Then  $\bar{x}$  satisfies (1.15) for some positive  $\delta > 0$ . Suppose  $\bar{x} \notin R^*$ , so  $\exists y \in R$ ,  $y \neq \bar{x}$ , s.t.  $f(y) < f(\bar{x})$ . Then  $\|y - \bar{x}\| \geq \delta$ . Choose  $z$  where

$$z = (1 - \lambda)x + \lambda y, \quad \lambda = \frac{\delta}{2\|y - \bar{x}\|}.$$

Then  $\|z - \bar{x}\| = \frac{\delta}{2} < \delta$ . Since  $R$  is convex by Th 1.11(a),  $z \in R$ . Since  $f$  is convex,

$$f(z) = f((1 - \lambda)\bar{x} + \lambda y) \leq (1 - \lambda)f(\bar{x}) + \lambda f(y) = f(\bar{x}) + \lambda(f(y) - f(\bar{x})) < f(\bar{x}).$$

which contradicts (1.15). Thus  $\bar{x} \in R^*$ . □

**Theorem 1.10.** Th 1.13 Suppose that (1.14) is a CP, where  $f$  is differentiable. Then  $\bar{x} \in R^* \iff \bar{x} \in R$ , and

$$\nabla f(\bar{x})^T(y - \bar{x}) \geq 0, \quad \forall y \in R. \tag{1.16}$$

*Proof.*  $\Leftarrow$

Suppose  $\bar{x} \in R$  and satisfies (1.16). If  $y \in R$ , then

$$f(y) \underset{\text{Th 1.9}}{\geq} f(\bar{x}) + \nabla f(\bar{x})^T(y - \bar{x}) \underset{(1.16)}{\geq} f(\bar{x}),$$

so  $\bar{x}$  is optimal.

$\implies$

Suppose  $\bar{x} \in R^*$ . Then  $\bar{x} \in R$ . Suppose (1.16) is false. Then  $\exists y \in R$  s.t.  $\nabla f(\bar{x})^T(y - \bar{x}) < 0$ . Consider  $z(\lambda) = (1 - \lambda)\bar{x} + \lambda y$ , where  $\lambda \in (-\varepsilon, 1 - \varepsilon)$  for some small  $\varepsilon > 0$ . By Th 1.11(a),  $R$  is convex, so  $z(\lambda) \in R$ . Now

$$\begin{aligned} \frac{d}{d\lambda} f(z(\lambda)) \Big|_{\lambda=0} & \underset{\text{CHAIN RULE}}{=} Df(z(\lambda))Dz(\lambda) = \nabla f(z(\lambda))^T(y - \bar{x}) \\ & = \nabla f(\bar{x})^T(y - \bar{x}) < 0 \end{aligned}$$

so if  $\lambda > 0$  is sufficiently small, then  $f(z(\lambda)) < f(z(0)) = f(\bar{x})$ . Thus  $\bar{x} \notin R^*$ , a contradiction. □

**Theorem 1.11.** Th 1.14 Suppose that (1.14) is a CP and is unconstrained, that is,  $m = 0$ . Let  $f$  be differentiable. Then  $\bar{x} \in R^* \iff \nabla f(\bar{x}) = 0$ .

*Proof.* ( $\implies$ ) Let  $\bar{x} \in R^*$ . Then Th 1.13  $\implies \nabla f(\bar{x})^T(y - \bar{x}) \geq 0, \forall y \in R$ . Since  $f$  is differentiable,  $\exists$  open neighborhood  $U$  of  $\bar{x}$ , s.t.  $\forall y \in U$ , we have  $y \in R$ . Let  $y = \bar{x} - t\nabla f(\bar{x})$  where  $t > 0$  is small, and  $y \in R$ . Then

$$0 \leq \nabla f(\bar{x})^T(y - \bar{x}) = -t\|\nabla f(\bar{x})\|^2,$$

where implies  $\nabla f(\bar{x}) = 0$ .

( $\impliedby$ ) Suppose  $\nabla f(\bar{x}) = 0$ . Clearly  $\bar{x} \in X_0$ , so  $\bar{x} \in R$ . Also,  $\nabla f(\bar{x})^T(y - \bar{x}) = 0, \forall y \in R$ . Thus TH 1.13  $\implies \bar{x}$  is optimal.  $\square$

### 1.2.1 Duality and the Lagrangian

Given problem (1.14). suppose the domain  $D \neq \emptyset$ . We define the **LAGRANGIAN**  $L : D \times \mathbb{R}^m \rightarrow \mathbb{R}$  by

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x)$$

**Definition 1.5.** The **LAGRANGE DUAL FUNCTION** is defined by

$$h(\lambda) = \inf_{x \in D} L(x, \lambda) = \inf_{x \in D} \left( f(x) + \sum_{i=1}^m \lambda_i g_i(x) \right)$$

The **(LAGRANGE) DUAL PROBLEM** of (1.14) is

$$\begin{aligned} \max \quad & h(\lambda) \\ & \lambda \geq 0 \end{aligned} \tag{1.17}$$

where  $\lambda \in \mathbb{R}^m$  is a variable. Let  $d^*$  denote the optimal value of problem (1.17). If  $h(\bar{\lambda}) = d^*$  for some  $\bar{\lambda} \in \mathbb{R}^m, \lambda \geq 0$ , then  $d^*$  is **ATTAINED** (by  $\bar{\lambda}$ ), and  $\bar{\lambda}$  is an **OPTIMAL SOLUTION** of (1.17). We also call (1.14) the **PRIMAL PROBLEM**.

**Theorem 1.12.** Th 1.15 **WEAK DUALITY**

Suppose problem (1.14) has an optimal value  $p^*$ . Then  $\forall \lambda \in \mathbb{R}^m, \lambda \geq 0$ ,

$$p^* \geq h(\lambda)$$



Thus,  $p^* \geq d^*$ . This property is **WEAK DUALITY** between problems (1.14) and (1.17).

*Proof.* The result holds if  $h(\lambda) = -\infty$  or  $p^* = \infty$ , so assume otherwise. Then  $R \neq \emptyset$ . Let  $y \in R$  in (1.14), so  $g_i(y) \leq 0, \forall 1 \leq i \leq m$ . Then for  $\lambda \in \mathbb{R}^m, \lambda \geq 0$ ,

$$h(\lambda) = \inf_{x \in D} L(x, \lambda) \leq L(y, \lambda) = f(y) + \sum_{i=1}^m \lambda_i g_i(y) \leq f(y).$$

Since  $h(\lambda) \leq f(y), \forall y \in R$ , we have  $h(\lambda) \leq p^*$ . □

If  $p^* = d^*$ , then we have **STRONG DUALITY** between problems (1.14) and (1.17). We do not always have strong duality, even if (1.14) is a CP.

We can show that the Lagrange dual problem of the LP (1.4) is the dual LP (1.7)<sup>2</sup>.

### 1.2.2 KT conditions

We will see that under some conditions, we have a necessary and sufficient condition for **STRONG DUALITY** between (1.14) and (1.17).

#### **Theorem 1.13.** TH 1.16 **KUHN-TUCKER THEOREM**

We have the primal problem (1.14) and its dual problem (1.17).

(a) Suppose that the optimal values  $p^*$  of (1.14) and  $d^*$  of (1.17) are both attained, and we have strong duality  $p^* = d^*$ . Let  $\bar{x}$  and  $\bar{\lambda}$  be optimal solutions. Then

$$g_i(\bar{x}) \leq 0, \quad 1 \leq i \leq m \quad \text{Primal feasible} \quad (1.18 \text{ a})$$

$$\bar{\lambda}_i \geq 0, \quad 1 \leq i \leq m \quad \text{Dual feasible} \quad (1.18 \text{ b}) \quad (1.18)$$

$$\bar{\lambda}_i g_i(\bar{x}) = 0 \quad 1 \leq i \leq m \quad \text{Complementary slacks} \quad (1.18 \text{ c})$$

If in addition,  $f$  and  $g_i$  ( $1 \leq i \leq m$ ) are differentiable (so the domains  $X_i$  ( $0 \leq i \leq m$ ) are open), then

$$\nabla f(\bar{x}) + \sum_{i=1}^m \bar{\lambda}_i \nabla g_i(\bar{x}) = 0 \quad \text{Stationary condition} \quad (1.18 \text{ d})$$

(b) Conversely, suppose that (1.14) is a CP, and  $f$  and  $g_i$  ( $1 \leq i \leq m$ ) are differentiable. If  $\bar{x}$  and  $\bar{\lambda}$  satisfy the conditions (1.18 a-d), then  $\bar{x}$  and  $\bar{\lambda}_i$  are optimal solutions yielding strong duality  $p^* = d^*$ .

The condition (1.18 a-d) are the KT conditions.

---

<sup>2</sup>HW 2 problem

*Proof.* (a) Clearly (1.18 a, b) hold. Now,

$$\begin{aligned} f(\bar{x}) &= h(\bar{\lambda}) = \inf_{x \in D} L(x, \bar{\lambda}) \leq L(x, \bar{\lambda}) \\ &= f(\bar{x}) + \sum_{i=1}^m \bar{\lambda}_i g_i(\bar{x}) \leq f(\bar{x}) \end{aligned}$$

so equality holds throughout. We have  $\sum_{i=1}^m \bar{\lambda}_i g_i(\bar{x}) = 0$  and (1.18 a-b)  $\Rightarrow \bar{\lambda}_i h_i(\bar{x}_i) \leq 0$  ( $1 \leq i \leq m$ ), so (1.18 c) holds.

Also  $\bar{x}$  minimises  $L(x, \bar{\lambda})$ . Suppose  $f$  and  $g_i$  ( $1 \leq i \leq m$ ) are differentiable. Then  $\nabla_x L(\bar{x}, \bar{\lambda}) = 0$ , which is (1.18 d).

(b) Since  $L(x, \bar{\lambda})$  is a sum of convex functions in  $x$ ,  $L(x, \bar{\lambda})$  is convex in  $x^3$ . Since (1.18 d) says  $\nabla_x L(\bar{x}, \bar{\lambda}) = 0$ , Th 1.14  $\Rightarrow \bar{x}$  minimises  $L(x, \bar{\lambda})$ . Thus,

$$h(\bar{\lambda}) = L(\bar{x}, \bar{\lambda}) = f(\bar{x}) + \sum_{i=1}^m \bar{\lambda}_i g_i(\bar{x}) \stackrel{(1.18 \text{ c})}{=} f(\bar{x}).$$

Thus  $\bar{x}$  and  $\bar{\lambda}$  are optimal solutions. and  $p^* = d^*$ . □

**Remark 1.9.**  $\nabla_x$  means  $\nabla$  is taken w.r.t  $x$ .

**Example 1.6.** Solve the CP

$$\begin{aligned} \min \quad & f(x_1, x_2) = (x_1 - 6)^2 + (x_2 - 5)^2 \\ \begin{cases} x_1^2 - 2x_1 + x_2 & \leq 8 \\ x_1, x_2 & \geq 0 \end{cases} \end{aligned} \quad (1.19)$$

By Th 1.16, the solution to (1.19) is achieved by any  $\bar{x}, \bar{\lambda}$  satisfying the KT conditions (1.18). We have:

$$\begin{cases} g_1(x_1, x_2) = x_1^2 - 2x_1 + x_2 - 8 \leq 0 & (1.20 \text{ a}) \\ g_2(x_1, x_2) = -x_1 \leq 0 & (1.20 \text{ a}') \\ g_3(x_1, x_2) = -x_2 \leq 0 & (1.20 \text{ a'') } \end{cases} \quad (1.20)$$

$$\lambda_1, \lambda_2, \lambda_3 \geq 0 \quad (1.20 \text{ b})$$

$$\lambda_1(x_1^2 - 2x_1 + x_2 - 8) = \lambda_2 x_1 = \lambda_3 x_2 = 0 \quad (1.20 \text{ c})$$

$$\begin{pmatrix} 2(x_1 - 6) + \lambda_1(2x_2 - 2) - \lambda_2 \\ 2(x_2 - 5) + \lambda_1 - \lambda_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (1.20 \text{ d})$$

Solving this, we have  $\bar{x}_1 = \frac{3+\sqrt{11}}{2}$ ,  $\bar{x}_2 = \frac{12-\sqrt{11}}{2}$ ,  $\lambda_1 = \sqrt{11} - 2$ ,  $\lambda_2 = \lambda_3 = 0$ . The minimum

---

<sup>3</sup>HW 2

value of  $f$  is:

$$(\bar{x}_1 - 6)^2 + (\bar{x}_2 - 5)^2 = \left( \frac{-9 + \sqrt{11}}{2} \right)^2 + \left( \frac{2 - \sqrt{11}}{2} \right)^2 = \frac{107 - 22\sqrt{11}}{4}$$

The detailed calculations is following:

Case 1.  $\lambda_1 = 0$ .

By (1.20 d), we have  $2(x_1 - 6) = \lambda_2$  and  $2(x_2 - 5) = \lambda_3$ . Then by (1.20 b, c), we have  $x_1 = 6, \lambda_2 = 0$ , and  $x_2 = 5, \lambda_3 = 0$ . But then  $g_1(6, 5) = 21$ , and (1.20a) does not hold. Thus we cannot have the case  $\lambda_1 = 0$ .

Case 2.  $\lambda_1 > 0$ .

By (1.20 c), we have  $x_1^2 - 2x_1 + x_2 - 8 = 0$ . If  $x_1 = 0$ , then (1.20d) gives  $2\lambda_1 + \lambda_2 = -12$ , which contradicts (1.20 b). So  $x_1 > 0$ , and (1.20c) implies  $\lambda_2 = 0$ . Then if  $x_2 = 0$ , we have  $0 = x_1^2 - 2x_1 - 8 = (x_1 - 4)(x_1 + 2)$ , so  $x_1 = 4$ . By (1.20d), we have  $-4 + 6\lambda_1 = 0$  and  $-10 + \lambda_1 - \lambda_3 = 0$ , which gives  $\lambda_1 = \frac{2}{3}$ , and  $\lambda_3 = -10 + \frac{2}{3} < 0$ , contradicting (1.20b). So  $x_2 > 0$ , and (1.20c) implies  $\lambda_3 = 0$ . Now (1.20d) gives  $2(x_1 - 6) + \lambda_1(2x_1 - 2) = 0$  and  $2(x_2 - 5) + \lambda_1 = 0$ . Eliminating  $\lambda_1$  and using  $x_2 = 8 - x_1^2 + 2x_1$ , we have  $x_1(2x_1^2 - 6x_1 - 1) = 0$ . Solving the quadratic gives  $\bar{x}_1 = \frac{3+\sqrt{11}}{2}$ , and then easy calculations give  $\bar{x}_2 = \frac{12-\sqrt{11}}{2}$  and  $\bar{\lambda}_1 = \sqrt{11} - 2$ . We already know that  $\bar{\lambda}_2 = \bar{\lambda}_3 = 0$ . This is the unique optimal solution, and the optimal value of  $f$  is

$$(\bar{x}_1 - 6)^2 + (\bar{x}_2 - 5)^2 = \left( \frac{-9 + \sqrt{11}}{2} \right)^2 + \left( \frac{2 - \sqrt{11}}{2} \right)^2 = \frac{107 - 22\sqrt{11}}{4}.$$

### 1.2.3 Algorithm Methods

In general, problem (1.14) can't be solved exactly. We consider algorithm methods we can give approximate solutions to (1.14).

We first consider the case when (1.14) is an unconstrained CP. That is:

$$\min f(x) \tag{1.21}$$

where  $f : X \rightarrow \mathbb{R}$  is a convex function, for some convex  $X \subset \mathbb{R}^n$ . Suppose  $f$  is twice continuously differentiable, so  $X$  is open, and (1.21) has an optimal solution  $\bar{x}$ . Suppose  $x^{(0)} \in X$  and  $x^{(0)} \notin R^*$ . We iteratively find  $x^{(0)}, x^{(1)}, x^{(2)}, \dots \in X$ , s.t.  $f(x^{(k)}) \rightarrow p^*$  as  $k \rightarrow \infty$ . Such  $\{x^{(k)}\}_{k=0}^\infty$  is a **MINIMISING SEQUENCE**. We end the algorithm when some form of accuracy is achieved. The set

$$S = \{x \in X : f(x) \leq f(x^{(n)})\}$$

is convex<sup>4</sup>. Suppose  $S$  is compact. We want to find the  $x^{(k)}$  s.t.

$$f(x^{(0)}) > f(x^{(1)}) > \dots > p^*$$

Such an algorithm is a DESCENT METHOD. We have  $x^{(k)} \in S, \forall k > 0$ . Each iteration involves a direction  $z^{(k)} \in \mathbb{R}^n$  and a step size  $t^{(k)} \geq 0$ . Thus

$$x^{(k+1)} = x^{(k)} + t^{(k)} z^{(k)},$$

or

$$x^+ = x + tz$$

if we want to focus on one iteration.

Th 1.8  $\Rightarrow f$  is convex on the line segment  $L = \{x + \lambda z : \lambda \geq 0\}$ , and  $f \Big|_L$  attains its minimum in  $L$ . We usually choose

$$t \in \{\lambda \geq 0 : f(x + \lambda z) \text{ is minimised on } L \text{ at } \lambda\}. \quad (1.22)$$

We remark that there are other choices for  $t$ , since (1.22) is not always easy to compute.

We consider two possible choices for the direction  $z$ .

#### A. STEEPEST DESCENT METHOD

We choose  $z = -\nabla f(x)$ . Th 1.14  $\Rightarrow \nabla f(x) \neq 0$ , since  $x \notin R^*$ . For small  $\lambda > 0$ ,

$$\begin{aligned} f(x^*) &= f(x + tz) \leq f(x + \lambda z) = f(x) + \lambda \nabla f(x)^T z + o(\lambda) \\ &= f(x) - \lambda \|\nabla f(x)\|^2 + o(\lambda) < f(x), \end{aligned}$$

where  $o(\lambda)$  means that  $\frac{o(\lambda)}{\lambda} \rightarrow 0$  as  $\lambda \rightarrow 0$ . So  $z = -\nabla f(x)$  is a suitable choice.

#### B. NEWTON'S METHOD

Assume  $\nabla^2 f(x) > 0, \forall x \in S$ . Note that  $\nabla^2 f(x)$  is invertible. Taylor's theorem  $\Rightarrow$

$$f(x + v) \approx f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v = \hat{f}(x + v)$$

$\hat{f}(x + v)$  is a convex quadratic function in  $v$ , minimised at  $v = \nabla^2 f(x)^{-1} \nabla f(x)^5$ . We define the

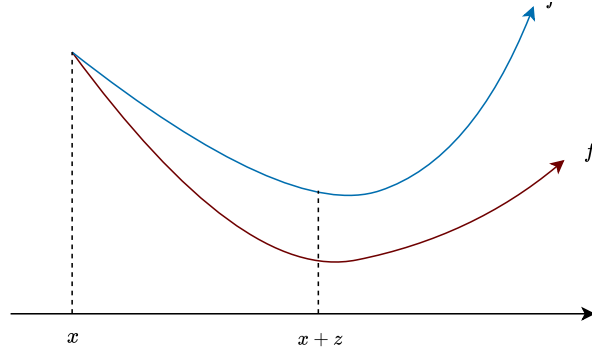
NEWTON STEP:

$$z = -\nabla^2 f(x)^{-1} \nabla f(x).$$

---

<sup>4</sup>In HW1

<sup>5</sup>HW 2



For small  $\lambda > 0$ ,

$$\begin{aligned} f(x^+) &= f(x + tz) \leq f(x + \lambda z) = f(x) + \lambda \nabla f(x)^T z + o(\lambda) \\ &= f(x) - \lambda \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) + o(\lambda) < f(x), \end{aligned}$$

since  $\nabla^2 f(x)^{-1} > 0$ . So  $z = -\nabla^2 f(x)^{-1} \nabla f(x)$  is a suitable choice.

$f(x) = o(g(x))$  means  $\frac{f(x)}{g(x)} \rightarrow 0$  as  $x \rightarrow c$  for some  $c$ . In most cases,  $c = \infty$  or  $c = 0$ .

Note: “ $o$  depends on  $c$ ”

Thus, we have the following algorithm for solving (1.21). Fix  $\varepsilon > 0$ . Suppose we found  $x^{(0)}, x^{(1)}, \dots, x^{(k)} \in S$ . We find  $z$  using  $A$  or  $B$  above. A stopping criteria can be  $\|z^{(k)}\| \leq \varepsilon$ . If  $\|z^{(k)}\| \geq \varepsilon$ , thus we find  $x^{(k+1)}$ , and iterate.

**Example 1.7.** Consider

$$\min f(x_1, x_2) = x_1^4 + 8x_2^4$$

where  $(x_1, x_2) \in \mathbb{R}^2$ . We have  $p^* = 0$ , where  $(x_1, x_2) = (0, 0)$ . Suppose we use the above algorithm with  $\varepsilon = 0.1$   $x^{(0)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ . For steepest descent method, we have

$$\nabla f(x) = \begin{pmatrix} 4x_1^3 \\ 32x_2^3 \end{pmatrix}, z^{(0)} = -\nabla f(x^{(0)}) = -\begin{pmatrix} 32 \\ 32 \end{pmatrix}$$

$$f(2 - 32\lambda, 1 - 32\lambda) = (2 - 32\lambda)^4 + 8(1 - 32\lambda)^4$$

$$\frac{df}{d\lambda} = -128(2 - 32\lambda)^3 - 1024(1 - 32\lambda)^3$$

Solving  $\frac{df}{d\lambda} = 0 \Rightarrow t^{(0)} = \frac{1}{24}$ . Thus,  $x^{(1)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} - \frac{1}{24} \begin{pmatrix} 32 \\ 32 \end{pmatrix} = \begin{pmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{pmatrix}$ . We have

$z^{(1)} = -\nabla f(x^{(1)}) = \begin{pmatrix} -\frac{32}{27} \\ \frac{32}{27} \end{pmatrix}$  and  $\|z^{(1)}\| = \frac{32}{27}\sqrt{2} > \varepsilon$ . We iterate this until we find  $z^{(k)}$  with  $\|z^{(k)}\| < \varepsilon = 0.1$ .

For Newton's method, we have:

$$\nabla^2 f(x) = \begin{pmatrix} 12x_1^2 & 0 \\ 0 & 96x_2^2 \end{pmatrix}, \quad \nabla^2 f(x)^{-1} = \begin{pmatrix} \frac{1}{12x_1^2} & 0 \\ 0 & \frac{1}{96x_2^2} \end{pmatrix}$$

$$-\nabla^2 f(x)^{-1} \nabla f(x) = -\begin{pmatrix} \frac{1}{12x_1^2} & 0 \\ 0 & \frac{1}{96x_2^2} \end{pmatrix} \begin{pmatrix} 4x_1^3 \\ 32x_2^3 \end{pmatrix} = -\begin{pmatrix} \frac{x_1}{3} \\ \frac{x_2}{3} \end{pmatrix}, \quad z^{(0)} = -\begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$$

Thus  $f(2 - \frac{2}{3}\lambda, 1 - \frac{1}{3}\lambda) = (2 - \frac{2}{3}\lambda)^4 + 8(1 - \frac{1}{3}\lambda)^4 = 24(1 - \frac{1}{3}\lambda)^4$ , which is minimised at  $\lambda = 3$ . So  $t^{(0)} = 3 \implies x^{(1)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} - 3 \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ , and we terminate in one step.

### PENALTY FUNCTION METHOD

Again, consider problem (1.14), with  $f$  and  $g_i (1 \leq i \leq m)$  continuous. We introduce the penalty program, which is the unconstrained problem:

$$\min \quad q(c, x) = f(x) + cp(x) \tag{1.23}$$

over  $x \in \mathbb{R}^n$ , where  $c > 0$ , and  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  is the penalty function, s.t.  $p$  is continuous,  $p(x) \geq 0, \forall x \in \mathbb{R}^n$  and  $p(x) = 0 \iff x \in R$ . The penalty term  $cp(x)$  gives a cost if  $x$  violates the constraints of (1.14).

**Remark 1.10.** Recall: If  $f : X \rightarrow \mathbb{R}$ , where  $X \subset \mathbb{R}^n$ , then

$$\arg \min_{x \in X} f(x) = \{y \in X : f(y) \leq f(x), \forall x \in X\}.$$

**Lemma 1.1.** Let  $0 < c_1 < c_2 < \dots$  with  $c_k \rightarrow \infty$ . Let  $q(c, x) = f(x) + cp(x)$  as in (1.23). Let  $x_k \in \arg \min_{x \in \mathbb{R}^n} q(c_k, x)$ . Then  $\forall k \geq 1$ ,<sup>a</sup>

$$(a) q(c_k, x_k) \leq q(c_{k+1}, x_{k+1}).$$

$$(b) p(x_k) \geq p(x_{k+1})$$

$$(c) f(x_k) \leq f(x_{k+1})$$

$$(d) f(x^*) \geq q(c_k, x_k) \geq f(x_k), \quad \text{where } x^* \in R^* \text{ (of problem (1.14))}.$$

<sup>a</sup>Proof HW2

**Remark 1.11.** Recall: For  $S \subset \mathbb{R}^n$ , a limit point of  $S$  is  $x \in \mathbb{R}^n$  s.t.

$\forall$  open set  $U$  with  $x \in U$ ,  $\exists y \in S \setminus \{x\}$ , s.t.  $y \in U$ .

**Theorem 1.14.** With the same notations as in (1.23) and Lemma 1.17, let  $\bar{x}$  be a limit point of  $\{x_k\}_{k=1}^\infty$ . Then  $\bar{x} \in R^*$  in problem (1.14).

*Proof.* Take a sub-sequence  $x_{k_1}, x_{k_2}, \dots$  of  $\{x_k\}_{k=1}^\infty$  s.t.  $x_{k_j} \rightarrow \bar{x}$ <sup>6</sup>. Since  $f$  and  $p$  are continuous,

$$\begin{aligned} \bar{q} &= \lim_{k_j \rightarrow \infty} q(c_{k_j}, x_{k_j}) = \lim_{k_j \rightarrow \infty} f(x_{k_j}) + \lim_{k_j \rightarrow \infty} c_{k_j} p(x_{k_j}) \stackrel{\text{Th 1.17(d)}}{\leq} f(x^*) \\ \implies \bar{q} - f(\bar{x}) &= \lim_{k_j \rightarrow \infty} c_{k_j} p(x_{k_j}) \leq f(x^*) - f(\bar{x}) \end{aligned}$$

Thus  $\lim_{k_j \rightarrow \infty} c_{k_j} p(x_{k_j})$  is finite, and since  $c_{k_j} \rightarrow \infty$ , we have  $p(x_{k_j}) \rightarrow 0$ . Thus  $p(\bar{x}) = 0$ , and  $\bar{x} \in R$ . Moreover,

$$\begin{aligned} f(x^*) &\geq f(\bar{x}) + \lim_{k_j \rightarrow \infty} c_{k_j} p(x_{k_j}) \geq f(\bar{x}) \\ \implies f(x^*) &= f(\bar{x}), \quad \text{and } \bar{x} \in R^* \end{aligned}$$

□

A commonly used penalty function is

$$p(x) = \sum_{i=1}^m [\max(0, g_i(x))]^t, \quad \text{for some } t \geq 1. \quad (1.24)$$

Especially, the case  $t = 2$ .

Suppose (1.14) also has equality constraints:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \begin{cases} g_i(x) \leq 0, & 1 \leq i \leq m \\ h_i(x) = 0 & 1 \leq i \leq r \end{cases} \end{aligned} \quad (1.25)$$

where  $h_i : Y_i \rightarrow \mathbb{R}$ , with  $Y_i \subset \mathbb{R}^n$  the domain of  $h_i$ . We may rewrite (1.25) in the form of (1.14).

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \begin{cases} g_i(x) \leq 0, & 1 \leq i \leq m \\ h_i(x) \leq 0 & 1 \leq i \leq r \\ -h_i(x) \leq 0 & 1 \leq i \leq r \end{cases} \end{aligned}$$

---

<sup>6</sup>HW 2

and work from there. The penalty function (1.24) becomes

$$p(x) = \sum_{i=1}^m [\max(0, g_i(x))]^t + \sum_{i=1}^r |h_i(x)|^t \quad (1.26)$$

**Example 1.8.** Use the penalty function method to solve

$$\begin{aligned} \min \quad & -3x_1 + x_2 \\ & x_1^2 - 2x_2 \leq 4 \end{aligned}$$

From (1.23) and (1.26) with  $t = 2$ , we have

$$q(c_k, x) = \begin{cases} -3x_1 + x_2 & \text{if } x_1^2 - x_2 - 4 \leq 0 \\ -3x_1 + x_2 + c_k (x_1^2 - x_2 - 4)^2 & \text{if } x_1^2 - x_2 - 4 > 0 \end{cases}$$

We find  $x_k \in \arg \min_{x \in \mathbb{R}^n} q(c_k, x)$ .

- If  $x_1^2 - 2x_2 - 4 \leq 0$ , then  $-3x_1 + x_2 \leq \frac{1}{2}x_1^2 - 3x_2 - 2 = \frac{1}{2}(x_1 - 3)^2 \geq -\frac{13}{2}$ , with equality when  $(x_1, x_2) = (3, \frac{5}{2})$ . Minimum  $q(c_k, x) = -3 \cdot 3 + \frac{5}{2} = -\frac{13}{2}$
- If  $x_1^2 - 2x_2 - 4 > 0$ . We solve  $\nabla_x q(c_k, x) = 0$  (See Th 1.14). We have

$$\nabla_x q(c_k, x) = \begin{pmatrix} -3 + 4c_k x_1 (x_1^2 - 2x_2 - 4) \\ 1 - 4c_k (x_1^2 - 2x_1 - 4) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

We can find  $x_1 = 3$ ,  $x_2 = \frac{5}{2} - \frac{1}{8c_k}$ , and minimum  $q(c_k, x) = -\frac{13}{2} - \frac{1}{16c_k}$ . Thus  $x_k = (3, \frac{5}{2} - \frac{1}{8c_k}) \rightarrow (3, \frac{5}{2})$ .  
Th 1.18

$$\implies \bar{x} = (3, \frac{5}{2}) \in R^*, \text{ and } p^* = -3 \cdot 3 + \frac{5}{2} = -\frac{13}{2}$$

**Example 1.9.** Use the penalty program method to solve

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 + 4x_3^2 \\ & 1 - x_1 - x_2 - x_3 = 0 \end{aligned}$$

From (1.23) and (1.26) with  $t = 2$ , we have

$$q(c_k, x) = x_1^2 + 2x_2^2 + 4x_3^2 + c_k(1 - x_1 - x_2 - x_3)^2$$



To find  $x_k \in \arg \min_{x \in \mathbb{R}^n} q(c_k, x)$ , we solve  $\nabla_x q(c_k, x) = 0$ . We have:

$$\nabla_x q(c_k, x) = \begin{pmatrix} 2x_1 - 2c_k(1 - x_1 - x_2 - x_3) \\ 4x_2 - 2c_k(1 - x_1 - x_2 - x_3) \\ 8x_3 - 2c_k(1 - x_1 - x_2 - x_3) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Then  $x_1 = 2x_2 = 4x_3$ , and  $x_k = \left( \frac{4c_k}{4+7c_k}, \frac{2c_k}{4+7c_k}, \frac{c_k}{4+7c_k} \right)$ . Since  $x_k \rightarrow (\frac{4}{7}, \frac{2}{7}, \frac{1}{7})$ .  $\bar{x} = (\frac{4}{7}, \frac{2}{7}, \frac{1}{7})$ . We have  $p^* = \frac{4}{7}$ .

### Algorithm Method with linear constraints

Consider the CP

$$\begin{aligned} \min \quad & f(x) \\ & Ax \leq b \end{aligned} \tag{1.27}$$

where  $f : X_0 \rightarrow \mathbb{R}$  with  $X_0 \subset \mathbb{R}^n$  open,  $f$  convex and continuously differentiable, and  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $x \in \mathbb{R}^n$ . Let  $R = \{x \in \mathbb{R}^n : Ax \leq b\}$  and assume  $R \neq \emptyset$ . Note that  $R$  is convex. Let  $x^{(0)} \in R$  with  $x^{(0)} \notin R^*$ . Taylor's theorem  $\implies$

$$\begin{aligned} f(x) &\approx f(x^{(0)}) + \nabla f(x^{(0)})^T (x - x^{(0)}) \\ &= f(x^{(0)}) - \nabla f(x^{(0)})^T x^{(0)} + \nabla f(x^{(0)})^T x \end{aligned}$$

Thus we consider

$$\begin{aligned} \min \quad & f(x^{(0)})^T x \\ & Ax \leq b \end{aligned} \tag{1.28}$$

Let  $y^{(0)}$  be an optimal solution of (1.28), so  $\nabla f(x^{(0)})^T (y^{(0)} - x^{(0)}) \leq 0$ . If  $\nabla f(x^{(0)})^T (y^{(0)} - x^{(0)}) = 0$ , then  $\forall x \in R$ ,

$$\begin{aligned} f(x) &\underset{\text{Th 1.9}}{\geq} f(x^{(0)}) + \nabla f(x^{(0)})^T (x - x^{(0)}) \\ &\geq f(x^{(0)}) + \nabla f(x^{(0)})^T (y^{(0)} - x^{(0)}) \\ &= f(x^{(0)}), \end{aligned}$$

so  $x^{(0)} \in R^*$ , a contradiction. Thus  $\nabla f(x^{(0)})^T (y^{(0)} - x^{(0)}) < 0$ . Now  $R$  convex  $\implies (1 - \lambda)x^{(0)} + \lambda y^{(0)} \in R$ ,

$\forall \lambda \in [0, 1]$ . Let

$$\lambda^{(0)} \in \arg \min_{\lambda \in [0, 1]} f((1 - \lambda)x^{(0)} + \lambda y^{(0)}),$$

$$x^{(1)} = (1 - \lambda^{(0)})x^{(0)} + \lambda^{(0)}y^{(0)}$$

Then for  $\lambda \in (0, 1)$  small,

$$\begin{aligned} f(x^{(1)}) &= f((1 - \lambda^{(0)})x^{(0)} + \lambda^{(0)}y^{(0)}) \\ &\leq f((1 - \lambda)x^{(0)} + \lambda y^{(0)}) \\ &= f(x^{(0)} + \lambda(y^{(0)} - x^{(0)})) \\ &= f(x^{(0)} + \lambda \nabla f(x^{(0)})^T (y^{(0)} - x^{(0)}) + o(\lambda)) \\ &< f(x^{(0)}) \end{aligned}$$

Repeating, we obtain a minimising sequence  $x^{(0)}, x^{(1)}, x^{(2)}, \dots \in R$  for problem (1.27). A stopping criteria can be when we have  $k$  s.t.  $|\nabla f(x^{(k)})^T (y^{(k)} - x^{(k)})| < \varepsilon$ , for some given  $\varepsilon > 0$ .

**Example 1.10.** Consider

$$\begin{aligned} \min \quad & f(x_1, x_2) = 4x_1^2 + (x_2 - 2)^2 \\ & \begin{cases} -2 \leq x_1 \leq 2 \\ -1 \leq x_2 \leq 1 \end{cases} \end{aligned}$$

Let  $x^{(0)} = (-2, -1)^T$ . We have  $\nabla f(x) = (8x_1, 2(x_2 - 2))^T$ , and

$$\begin{aligned} \min \quad & \nabla f(x^{(0)})^T x = -16x_1 - 6x_2 \\ & \begin{cases} -2 \leq x_1 \leq 2 \\ -1 \leq x_2 \leq 1 \end{cases} \end{aligned}$$

has optimal solution  $y^{(0)} = (2, 1)^T$ . Now

$$f((1 - \lambda)x^{(0)} + \lambda y^{(0)}) = f(4\lambda - 2, 2\lambda - 1) = 68\lambda^2 - 76\lambda + 25$$

Solving  $\frac{df}{d\lambda} = 0$  gives  $\lambda^{(0)} = \frac{19}{34}$ . Thus

$$x^{(1)} = \frac{15}{34} \begin{pmatrix} -2 \\ -1 \end{pmatrix} + \frac{19}{34} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{4}{17} \\ \frac{2}{17} \end{pmatrix}$$

We may repeat this procedure to obtain a minimising sequence  $x^{(1)}, x^{(2)}, \dots$ .

## 1.3 Integer programming

Consider the LP

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{aligned} \quad (1.5)$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ ,  $x \in \mathbb{R}^n$ , with optimal value  $p^*$ . In many applications, we will be interested in the restriction  $x \in \mathbb{Z}_{\geq 0}^n$ . For example, suppose a company sells  $n$  different types of products, where  $x_i$  is the number of units that product of type  $i$  is sold. The company wants to maximise the profit, with some constraints are sold. Then we would be interested in  $x_i \in \mathbb{Z}_{\geq 0}$ .

Thus, we consider the problem.

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & \begin{cases} Ax \leq b \\ x \in \mathbb{Z}_{\geq 0}^n \end{cases} \end{aligned} \quad (1.29)$$

Let  $q^*$  be the optimal value. (1.29) is an integer programming problem (IP). The case

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & \begin{cases} Ax \leq b \\ x_i \in \{0, 1\}, \quad 1 \leq i \leq n \end{cases} \end{aligned} \quad (1.30)$$

is a **BINARY INTEGER PROGRAMMING PROBLEM (BIP)**. We say that (1.5) is that LP relaxation of (1.29).

Note that  $q^* \leq p^* \iff$  (1.5) has an optimal solution in  $\mathbb{Z}_{\geq 0}^n$ .

We shall consider two methods for solving (1.29) and (1.30).

### A. Branch and bound method

**Idea:** We divide the IP (1.29) into several subproblems, and work towards finding the optimal value  $q^*$ .

**Example 1.11.** Solve

$$\max \quad f = -x_1 + 4x_2 \quad (1.31)$$

$$\begin{cases} -5x_1 + 10x_2 \leq 11 \\ 5x_1 + 10x_2 \leq 49 \\ x_1 \leq 5 \\ x_1, x_2 \in \mathbb{Z}_{\geq 0} \end{cases} \quad (1.31 \text{ a})$$

The optimal solution of (1.31 a) is  $(2, 2)$ , and the optimal value is  $q^* = 6$ . See the handout ?? for full details.

We may also consider the version of the IP (1.29) where some of the  $x_i$  are integers.

### B. Mixed integer programming problem

A **MIXED INTEGER PROGRAMMING PROBLEM (MIP)** has the form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & \begin{cases} Ax \leq b \\ x_i \in \mathbb{Z}_{\geq 0}, & i \in I \\ x_i \geq 0, & i \in \{1, 2, \dots, n\} \setminus I \end{cases} \end{aligned}$$

for some fixed  $I \subset \{1, 2, \dots, n\}$ .

See Example in handout ?? where (1.31 a) is modified to  $x_i \in \mathbb{Z}_{\geq 0}$ ,  $x_2 \geq 0$ .

### C. BIP problem

For BIP, we may also use the same method. The following method is similar, but more simple.

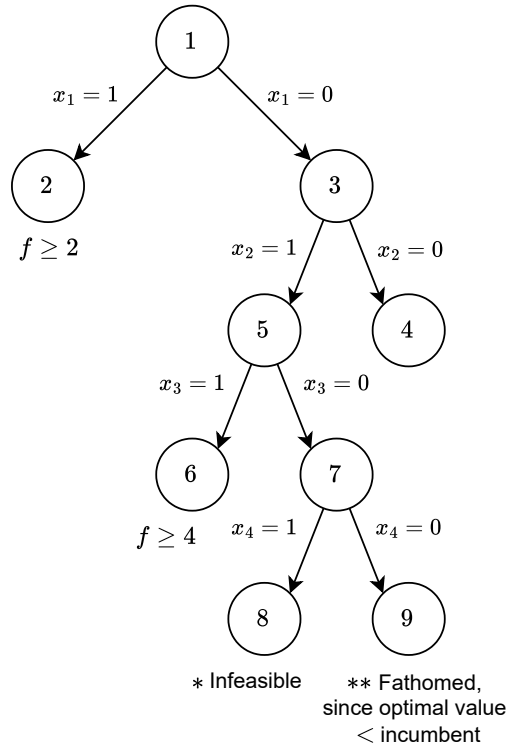
**Example 1.12.** Solve

$$\begin{aligned} \max \quad & f = -8x_1 - 2x_2 - 4x_3 - 7x_4 - 5x_5 + 10 \\ \text{s.t.} \quad & \begin{cases} -3x_1 - 3x_2 + x_3 + 2x_4 + 3x_5 \leq -2 \\ -5x_1 - 3x_2 - 2x_3 - x_4 + x_5 \leq -4 \\ x_i \in \{0, 1\}, & 1 \leq i \leq 5 \end{cases} \end{aligned}$$

Note that to maximise  $f$ , we would “prefer  $x_i = 0$  for as many  $i$  as possible”.

1. Consider  $x_1 = 0$  and  $x_1 = 1$ .
2. If  $x_1 = 1$ , then setting  $x_2 = x_3 = x_4 = x_5 = 0$ , the constraints are satisfied. We have an incumbent optimal value of 2.
3. Let  $x_1 = 0$ . Consider  $x_2 = 0$  and  $x_2 = 1$ .
4. If  $x_2 = 0$ , then first constraint can't be satisfied.
5. Let  $x_2 = 1$ . Consider  $x_3 = 0$  and  $x_3 = 1$ .
6. If  $x_3 = 1$ , then setting  $x_4 = x_5 = 0$ , the constraints are satisfied. We have a new incumbent optimal value of 4.
7. Let  $x_3 = 0$ . Consider  $x_4 = 0$  and  $x_4 = 1$ .
8. If  $x_4 = 1$ , then the objective value is  $\leq 1$ ,  $<$  current incumbent optimal value of 4.
9. Let  $x_4 = 0$ . We currently have  $(x_1, x_2, x_3, x_4) = (0, 1, 0, 0)$ . Second constraint is not satisfied.

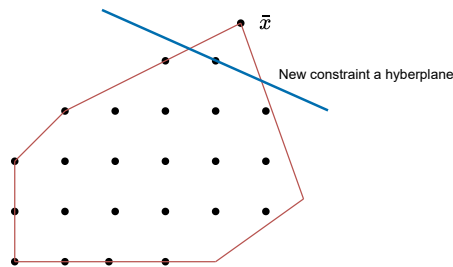
We conclude that the optimal solution is  $(0, 1, 1, 0, 0)$ , and the optimal value is 4. We have the following branching procedure:



Note: If the coefficient of some  $x_i$  in the objective function is positive, we may substitute  $x'_i = 1 - x_i$ .

### 1.3.1 B. Cutting plane method

Idea: Suppose that in the IP (1.29), the LP relaxation (1.4) has an optimal solution  $\bar{x} \notin \mathbb{Z}_{\geq 0}^n$ . We add a new constraint into (1.4) which defines a hyperplane. The hyperplane “cuts off”  $\bar{x}$  from the feasible set of (1.4) but none of the feasible integer points of (1.29).



We solve the new LP, and repeat the procedure until we find an optimal solution in  $\mathbb{Z}_{\geq 0}^n$  for some LP.

Here, we assume the entries of  $A, b, c$  in (1.29) are in  $\mathbb{Z}$ , or equivalently, in  $\mathbb{Q}$  (since we may clear fractions by multiplying by a suitable integer).

**Lemma 1.2.** Lemma 1.19 Let  $x_i \in \mathbb{Z}_{\geq 0}$ ,  $a_i \in \mathbb{R}$  for  $1 \leq i \leq n$ , and  $b \in \mathbb{R}$ . If  $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$ , then

$$\begin{aligned} \lfloor a_1 \rfloor x_1 + \cdots + \lfloor a_n \rfloor x_n &\leq \lfloor b \rfloor, \\ \text{and } (a_1 - \lfloor a_1 \rfloor)x_1 + \cdots + (a_n - \lfloor a_n \rfloor)x_n &\geq b - \lfloor b \rfloor. \end{aligned}$$

*Proof.* We have

$$\begin{aligned} b &= (a_1 - \lfloor a_1 \rfloor)x_1 + \cdots + (a_n - \lfloor a_n \rfloor)x_n + \lfloor a_1 \rfloor x_1 + \cdots + \lfloor a_n \rfloor x_n \\ &\geq \lfloor a_1 \rfloor x_1 + \cdots + \lfloor a_n \rfloor x_n. \end{aligned}$$

Since  $\lfloor a_1 \rfloor x_1 + \cdots + \lfloor a_n \rfloor x_n \in \mathbb{Z}$ , we have  $\lfloor a_1 \rfloor x_1 + \cdots + \lfloor a_n \rfloor x_n \leq \lfloor b \rfloor$ .

Multiplying by  $-1$ , and adding  $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$  gives  $(a_1 - \lfloor a_1 \rfloor)x_1 + \cdots + (a_n - \lfloor a_n \rfloor)x_n \geq b - \lfloor b \rfloor$ .  $\square$

**Example 1.13.** Use the cutting plane method to solve

$$\begin{aligned} \max \quad & f = 2x_1 + 3x_2 \\ \left\{ \begin{array}{ll} -2x_1 + 2x_2 & \leq 3 \\ x_1 + 2x_2 & \leq 6 \\ 4x_1 + 5x_2 & \leq 20 \\ x_1, x_2 & \in \mathbb{Z}_{\geq 0} \end{array} \right. \end{aligned} \tag{1.32}$$

Introduce slack variables  $s_1, s_2, s_3 \geq 0$  to the LP relaxation of (1.32). By simplex method, we have

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$f$	
-2	2	1	0	0	0	3
1	2	0	1	0	0	6
4	5	0	0	1	0	20
-2	-3	0	0	0	1	0

After some work, we have

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$f$	
0	1	0	$\frac{4}{3}$	$-\frac{1}{3}$	0	$\frac{4}{3}$
1	0	0	$-\frac{5}{3}$	$\frac{3}{2}$	0	$\frac{10}{3}$
0	0	1	-6	2	0	7
0	0	0	$\frac{2}{3}$	$\frac{1}{3}$	1	$\frac{32}{3}$

The final table says that we have not found an integer solution to the LP relaxation. The entries on the right of rows 1, 2, 4 are non-integers. Note that  $f \in \mathbb{Z}_{\geq 0}$ . We have

$$x_2 + \frac{4}{3}s_2 - \frac{1}{3}s_3 = \frac{4}{3} \quad \frac{1}{3}s_2 + \frac{2}{3}s_3 \geq \frac{1}{3} \quad (1)$$

$$x_1 - \frac{5}{3}s_2 + \frac{2}{3}s_3 = \frac{10}{3} \quad \xrightarrow{\text{Lemma 1.19}} \quad \frac{1}{3}s_2 + \frac{2}{3}s_3 \geq \frac{1}{3} \quad (2)$$

$$\frac{2}{3}s_2 + \frac{1}{3}s_3 + f = \frac{32}{3} \quad \frac{2}{3}s_2 + \frac{1}{3}s_3 \geq \frac{2}{3} \quad (3)$$

Each of (1), (2), (3) is a Gomory fractional cut. We see that (1) and (2) are the same. Note that (3)  $\implies$  (1), (2), since (3)  $\implies \frac{1}{3}s_2 + \frac{1}{6}s_3 \geq \frac{1}{3}$ , which  $\implies$  (1), (2). Now

$$(3) \iff \frac{2}{3}(6 - x_1 - 2x_2) + \frac{1}{3}(20 - 4x_1 - 5x_2) \geq \frac{2}{3} \iff 2x_1 + 3x_2 \leq 10$$

Now we solve a new LP by adding the constraint (3), which we write as  $-\frac{2}{3}s_2 - \frac{1}{3}s_3 + s_4 = -\frac{2}{3}$  for a slack variable  $s_4 \geq 0$ . We use the dual simplex method:

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	
$x_2$	0	1	0	$\frac{4}{3}$	$-\frac{1}{3}$	0	$\frac{4}{3}$
$x_1$	1	0	0	$-\frac{5}{3}$	$\frac{2}{3}$	0	$\frac{10}{3}$
$s_1$	0	0	1	-6	2	0	-7
$s_4$	0	0	0	$-\frac{2}{3}$	$-\frac{1}{3}$ (pivot)	1	$-\frac{2}{3}$
$-f$	0	0	0	$-\frac{2}{3}$	$-\frac{1}{3}$	0	$-\frac{32}{3}$

Then we have:

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	
$x_2$	0	1	0	2	0	-1	2
$x_1$	1	0	0	-3	0	2	2
$s_1$	0	0	1	-10	0	6	3
$s_3$	0	0	0	2	1	-3	2
$-f$	0	0	0	0	0	-1	-10

We find the integer optimal solution  $(x_1, x_2) = (2, 2)$  for (1.32). The optimal values is  $f = 10$ .

**Remark 1.12.** (1), (2)  $\implies 3x_1 + 4x_2 \leq 15$ , which gives a weaker cutting line that  $2x_1 + 3x_2 \leq 10$  from (3). In general, we may obtain several cutting planes in dimension  $n$  which are incomparable.

If we did not find an integer optimal solution, we may repeat the whole procedure.

## 1.4 Dynamic Programming

A DYNAMIC PROGRAMMING (DP) PROBLEM can be described as a problem that can be solved by solving a sequence of subproblems. DPs occur in many areas of OR, not just in optimisation theory. We consider some important example.

### 1.4.1 A. Knapsack Problem

We would like to prepare a knapsack of items, such as water bottles, apples, etc, for a camping trip. There are  $n$  types of items, and each item of type  $i$  has weight  $a_i$  and a value of importance, or profit  $c_i$  ( $1 \leq i \leq n$ ). The knapsack can hold a total weight of  $b$ .

Thus we want to solve the DP

$$\begin{aligned} \max \quad & c^T x \\ \left\{ \begin{array}{l} a^T x \leq b \\ x \in \mathbb{Z}_{\geq 0}^n \end{array} \right. \end{aligned} \quad (1.33)$$

where  $c, a \in \mathbb{R}_{\geq 0}^n$ ,  $b \in \mathbb{R}$ . We have  $x_i$  is the number of items of type  $i$ .

We may solve (1.33) with a recurrence relation, as follows. For  $a \leq d \leq b$ ,  $1 \leq k \leq n$ . Let  $S_k(d)$  = maximum profits that can be achieved by item types  $k, k+1, \dots, n$ , if the knapsack has remaining weight  $d$ . We want  $S_1(b)$ .

Suppose we have assigned that values for  $x_1, \dots, x_{k-1}$ . We want to assign a value for  $x_k$ , and the knapsack has remaining weight  $d$ . Then we require  $0 \leq x_k \leq \lfloor \frac{d}{a_k} \rfloor$ . Adding the type  $k$  items increases the profit by  $c_k x_k$  and decreases the remaining weight  $a_k x_k$ . The best “future contribution” to the total profit in the knapsack is  $S_{k+1}(d - a_k x_k)$ . Thus

$$S_k(d) = \max \{ c_k x_k + S_{k+1}(d - a_k x_k) : 0 \leq x_k \leq \lfloor \frac{d}{a_k} \rfloor \}, \quad 1 \leq k \leq n-1 \quad (1.34)$$



**Example 1.14.** Solve the IP

$$\begin{aligned} \max \quad & 4x_1 + 5x_2 + 6x_3 \\ \text{s.t.} \quad & \begin{cases} 3x_1 + 4x_2 + 5x_3 \leq 10 \\ x_1, x_2, x_3 \in \mathbb{Z}_{\geq 0} \end{cases} \end{aligned}$$

*Solution.* Stage 3 Find  $S_3(d)$  for  $0 \leq d \leq 10$ .

$$S_3(d) = \begin{cases} 0, & 0 \leq d < 5, & (x_3 = 0) \\ 6, & 5 \leq d < 10, & (x_3 = 1) \\ 12, & d = 10, & (x_3 = 2) \end{cases}$$

Stage 2 Find  $S_2(d)$  for  $0 \leq d \leq 10$ . (1.34)  $\Rightarrow$

$$S_2(d) = \max\{5x_2 + S_3(d - 4x_2) : 0 \leq x_2 \leq \lfloor \frac{d}{4} \rfloor\}.$$

$0 \leq d < 4$ :  $x_2 = 0$ ,  $S_2(d) = S_3(d) = 0$ .

$4 \leq d < 8$ :  $x_2 \in \{0, 1\}$ ,  $S_2(d) = \max(S_3(d), 5 + S_3(d - 4)) = \max(S_3(d), 5) = \begin{cases} 5, & 4 \leq d < 5 \\ 6, & 5 \leq d < 8 \end{cases}$

$8 \leq d < 10$ :  $x_2 \in \{0, 1, 2\}$ ,

$$S_2(d) = \max(S_3(d), 5 + S_3(d - 4), 10 + S_3(d - 8)) = \begin{cases} \max(6, 5, 10) = 10, & 8 \leq d < 9, \\ \max(6, 11, 10) = 11, & 9 \leq d < 10, \\ \max(12, 11, 10) = 12, & d = 10. \end{cases}$$

Stage 1 Find  $S_1(10)$ . (1.34)  $\Rightarrow$

$$\begin{aligned} S_1(10) &= \max\{4x_1, S_2(10 - 3x_1) : 0 \leq x_1 \leq \lfloor \frac{10}{3} \rfloor\} \\ &= \max\{S_2(10), 4 + S_2(7), 8 + S_2(4), 12 + S_2(1)\} \\ &= \max\{12, 4 + 6, \boxed{8 + 5}, 12 + 0\} \\ &= 13. \end{aligned}$$

Optimal value = 13. Optimal solution:  $x_1 = 2$ .

$$\begin{aligned} S_2(4) &= \max\{0, \boxed{5}\} = 5 \Rightarrow x_2 = 1 \\ x_3 &= 0 \text{ from } 4x_1 + 5x_2 + 6x_3 = 13 \end{aligned}$$

□

We may more conveniently use a table method, which is a common approach in DP problems. The

values of  $S_1(d)$  ( $0 \leq d \leq 10$ ) are also found.

$\begin{matrix} d \\ k \end{matrix}$	0	1	2	3	4	5	6	7	8	9	10
3	<u>0</u> <sup>+5</sup>	0	0	0	0	6	6	6	6	6	12
2	0	0	0	0	<u>5</u> <sup>+8</sup>	6	6	6	10	<u>11</u>	12
1	0	0	0	4	5	6	8	9	10	12	<u>13</u>

Table 1.2: Example

Again, maximum =  $S_1(10) = 13$ . To calculate the values of the  $x_1$  achieving this maximum, we backtrack. See the entries in red. Note that if there is a tie during the back tracking, we may consider all possibilities to obtain all optimal solutions.

Table Method 1. Fill in the first row ( $k = n$ ):  $S_n(d) = c_n \lfloor \frac{d}{a_n} \rfloor$ .

2. Fill in each successive row ( $n > k \geq 1$ ) by using (1.34). Note that each row depends only on the entries in the row above.

We may add more constraints to problem (1.33). For two constraints, we have the ID:

$$\begin{aligned} \max \quad & c^T x \\ \text{subject to} \quad & \begin{cases} a^T x \leq b \\ p^T x \leq q \\ x \in \mathbb{Z}_{\geq 0}^n \end{cases} \end{aligned} \quad (1.35)$$

where  $c, a, q \in \mathbb{R}_{\geq 0}^n$ ,  $b, p \in \mathbb{R}_{\geq 0}$ .

We may interpret the second constraint as: Each item of type  $i$  ( $1 \leq i \leq n$ ) has volume  $p_i$ , and the knapsack can also hold a maximum volume of  $q$ .

For  $0 \leq d \leq b$ ,  $0 \leq r \leq q$ ,  $1 \leq k \leq n$ , let  $S_k(d, r)$  be the maximum profit from item types  $k, k+1, \dots, n$  with remaining weight  $d$  and volume  $r$ . We want  $S_1(b, q)$ . We may similarly obtain the following recurrence relation for (1.35)

$$S_k(d, r) = \max \{ c_k x_k + S_{k-1}(d - a_k x_k, r - p_k x_k) : 0 \leq x_k \leq \min \left( \left\lfloor \frac{d}{a_k} \right\rfloor, \left\lfloor \frac{r}{p_k} \right\rfloor \right), 1 \leq k \leq n-1 \} \quad (1.36)$$

**Example 1.15.** Solve the IP

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 \\ \text{subject to} \quad & \begin{cases} 3x_1 + 4x_2 \leq 12 \\ x_1 + 5x_2 \leq 10 \\ x_1, x_2 \in \mathbb{Z}_{\geq 0} \end{cases} \end{aligned} \quad (1.37)$$

*Solution.* Stage 2 For  $0 \leq d \leq 12$ ,  $0 \leq r \leq 10$ ,

$$S_2(d, r) = \begin{cases} 0, & \text{if } (d, r) \in ([0, 12] \times [0, 10]) \setminus ([4, 12] \times [5, 10]) \\ 3, & \text{if } (d, r) \in ([4, 12] \times [5, 10]) \setminus ([8, 12] \times \{10\}) \\ 6, & \text{if } (d, r) \in [9, 12] \times \{10\} \end{cases}$$

Stage 1 (1.36)  $\Rightarrow$

$$\begin{aligned} S_1(12, 10) &= \max\{2x_1 + S_2(12 - 3x_1, 10 - x_1) : 0 \leq x_1 \leq \min\left(\lfloor \frac{12}{3} \rfloor, \lfloor \frac{10}{1} \rfloor\right) = 4\} \\ &= \max\{S_2(12, 10), 2 + S_2(9, 9), 4 + S_2(6, 8), 6 + S_2(3, 7), 8 + S_2(0, 6)\} \\ &= \max(6, 2 + 3, 4 + 3, 6 + 0, 8 + 0) \\ &= 8 \end{aligned}$$

Optimal value = 8

Optimal solution:  $x_1 = 4, x_2 = 0$  for  $2x_1 + 3x_2 = 8$ .

We may also use a table method by considering  $k = 2$  and computing  $d$  versus  $r$ , then repeat for  $k = 1$ . □

### 1.4.2 B. 0-1 knapsack Problem

We have  $n$  items available, where item  $i$  has weight  $a_i$  and profit  $c_i$  ( $1 \leq i \leq n$ ). We want to place some of the items into a knapsack which holds a maximum weight  $b$ , and maximise the profit. We have the version of the knapsack problem where repetitions of the items are not allowed, or the 0-1 knapsack problem (a BIP).

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & \begin{cases} a^T x \leq b \\ x_i \in \{0, 1\} \quad 1 \leq i \leq n \end{cases} \end{aligned} \tag{1.38}$$

where  $c, a \in \mathbb{R}_{\geq 0}^n$ ,  $b \in \mathbb{R}_{\geq 0}$ . The recurrence relation for (1.38) is slightly different. For  $0 \leq d \leq b$ ,  $1 \leq k \leq n$ , let  $S_k(d)$  be the maximum profit from items  $k, k+1, \dots, n$  if the knapsack has remaining weight  $d$ .

Similar to (1.34), we have, for  $1 \leq k \leq n-1$ ,

$$S_k(d) = \begin{cases} S_{k+1}(d), & \text{if } 0 \leq d < a_k \\ \max(S_{k+1}(d), c_k + S_{k+1}(d - a_k)), & \text{if } a_k \leq d \leq b \end{cases} \tag{1.39}$$

**Example 1.16.** Solve the BIP

$$\begin{aligned} \max \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{s.t.} \quad & \begin{cases} 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ x_i \in \{0, 1\}, & 1 \leq i \leq 4 \end{cases} \end{aligned}$$

#### STAGE 4

$$S_4(d) = \begin{cases} 0, & \text{if } 0 \leq d < 3 \quad (x_4 = 0) \\ \boxed{4}, & \text{if } 3 \leq d \leq 14 \quad (x_4 = 1) \end{cases}$$

#### STAGE 3

(1.39)  $\Rightarrow$

$$S_3(d) = S_4(d) = \begin{cases} 0, & \text{if } 0 \leq d < 3 \quad (x_4 = 0) \\ 4, & \text{if } 3 \leq d \leq 14 \quad (x_4 = 1) \end{cases}$$

$$\begin{aligned} S_3(d) &= \max(S_4(d), \boxed{6 + S_4(d - 4)}) \\ &= \begin{cases} \max(4, 6 + 0) = 6, & \text{if } 4 \leq d < 7 \\ \max(4, \boxed{6 + 4}) = 10, & \text{if } 7 \leq d \leq 14 \end{cases} \end{aligned}$$

#### STAGE 2

$$\begin{aligned} S_2(d) = S_3(d) &= \begin{cases} 0, & \text{if } 0 \leq d < 3 \\ 4, & \text{if } 3 \leq d < 7 \\ 6, & \text{if } 7 \leq d < 10 \end{cases} \\ S_2(d) &= \max(S_3(d), \boxed{11 + S_3(d - 7)}) \\ &= \begin{cases} \max(10, 11 + 0) = 11, & \text{if } 7 \leq d < 10 \\ \max(10, 11 + 4) = 14, & \text{if } 10 \leq d < 11 \\ \max(10, 11 + 6) = 17, & \text{if } 11 \leq d \leq 14 \\ \max(10, \boxed{11 + 10}) = 21, & \text{if } d = 14 \end{cases} \end{aligned}$$

#### STAGE 1

$$(1.39) \Rightarrow S_1(14) = \max(\boxed{S_2(14)}, 8 + S_2(14 - 5)) = \max(\boxed{21}, 8 + 11) = 21$$

Optimal value = 21. Optimal solution:  $(x_1, x_2, x_3, x_4) = (0, 1, 1, 1)$ .

#### TABLE METHOD

$\begin{smallmatrix} d \\ k \end{smallmatrix}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Optimal solution
4	<span style="border: 1px solid black;">0</span>	0	0	<span style="border: 1px solid black;">4</span>	4	4	4	4	4	4	4	4	4	4	4	$x_4 = 1$
3	0	0	0	4	6	6	6	<span style="border: 1px solid black;">10</span>	10	10	10	10	10	10	10	$x_3 = 1$
2	0	0	0	4	6	6	6	11	11	11	15	17	17	17	<span style="border: 1px solid black;">21</span>	$x_2 = 1$
1	0	0	0	4	6	8	8	11	12	14	15	17	19	19	<span style="border: 1px solid black;">21</span>	$x_1 = 0$

Table 1.3: Table Method

As before, we may generalise (1.38) by having two or more constraints.

### 1.4.3 C. Resource allocation problem

Suppose we have a total budget (or resource)  $b \in \mathbb{Z}_{\geq 0}$ , to be spent on  $n$  types of activities. If  $x_i \in \mathbb{Z}_{\geq 0}$  units of activity type  $i$  are performed, there is a profit of  $f_i(x_i)$ , where  $f_i : \{0, 1, \dots, b\} \rightarrow \mathbb{R}_{\geq 0}$  is a non-decreasing function with  $f_i(0) = 0$ ,  $\forall 1 \leq i \leq n$ .

We want to solve the IP:

$$\begin{aligned} \max \quad & \sum_{i=1}^n f_i(x_i) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^n x_i = b \\ x_i \in \mathbb{Z}_{\geq 0} \end{cases} \end{aligned} \quad (1.40)$$

For  $0 \leq d \leq b$ ,  $1 \leq k \leq n$ , let  $S_k(d)$  be the maximum profit from activities of type  $k, k+1, \dots, n$ , with a budget of  $d$  remaining. We want  $S_1(b)$ . Similarly, we have the following recurrence relation for (1.40):

$$S_k(d) = \max\{f_k(x_k) + S_{k+1}(d - x_k) : 0 \leq x_k \leq d\}, \quad 1 \leq k \leq n-1 \quad (1.41)$$

Note that  $S_n(d) = f_n(d)$  for  $d = 1, 2, \dots, b$ , and  $S_k(0) = 0$  for  $k = 1, 2, \dots, n$ .

**Example 1.17.** Suppose 4 doctors are assigned to 3 hospitals. For  $i = 1, 2, 3$  and  $y = 1, 2, 3, 4$ , let  $f_i(y)$  be the number of patients that  $y$  doctors can serve per hour in hospital  $i$  given by the table

$y = \text{Number pf doctors}$	$f_1(y)$	$f_2(y)$	$f_3(y)$
0	0	0	0
1	4	2	5
2	6	4	7
3	9	7	8
4	10	11	9

Table 1.4: number of patients that  $y$  doctors can serve per hour

What is the maximum umber of patients that can be served per hour, across 3 hospitals?

We have problem (1.40) where  $b = 4$ , and  $x_1 =$  number of doctors assigned to hospital  $i$ .

### STAGE 3

$$S_3(d) = f_3(d), \quad 0 \leq d \leq 4$$

### STAGE 2

$$S_2(0) = 0$$

$$(1.41) \implies S_2(d) = \max\{f_2(x_2) + S_3(d - x_2) : 0 \leq x_2 \leq d\}$$

$$S_2(1) = \max\{0 + S_3(1), 2 + f_3(0)\} = \max\{5, 2\} = 5$$

$$S_2(2) = \max\{0 + S_3(2), 2 + S_3(1), 4 + S_3(0)\} = \max\{7, 7, 4\} = 7$$

$$S_2(3) = \max\{0 + S_3(3), 2 + S_3(2), 4 + S_3(1), 7 + S_3(0)\} = \{8, 9, 9, 7\} = 9$$

$$S_2(4) = \max\{0 + S_3(4), 2 + S_3(3), 4 + S_3(2), 7 + S_3(1), 11 + S_3(0)\} = \{9, 10, 11, 12, 11\} = 12$$

### STAGE 1

$$(1.41) = S_1(4) = \max\{f_1(x_1) + S_2(4 - x_1) : 0 \leq x_1 \leq 4\}$$

$$= \max\{0 + S_2(4), 4 + S_2(3), 6 + S_2(2), 9 + S_2(1), 10 + S_2(0)\} = \max\{12, 13, 13, 14, 10\} = 14$$

Optimal value = 14. Optimal solution :  $(x_1, x_2, x_3) = (3, 0, 1)$ . See the circled entries in the above table.

### TABLE METHOD

$\begin{array}{c} d \\ \backslash \\ k \end{array}$	0	1	2	3	4	Optimal solution
3	0	5	7	8	9	$x_3 = 1$
2	0	5	7	9	12	$x_2 = 0$
1	0	5	9	11	14	$x_1 = 3$

Table 1.5: Table Method





# Chapter 2

## Game Theory

### 2.1 Introduction

Suppose two players Alice and Bob play a game. The set of choices, called STRATEGIES, available to Alice and Bob, are  $S_1 = \{\alpha_1, \dots, \alpha_m\}$  and  $S_2 = \{\beta_1, \dots, \beta_n\}$ . If Alice and Bob choose  $\alpha_i$  and  $\beta_j$ , then they win  $a_{ij}$  and  $b_{ij}$  respectively. The matrices  $A, B \in \mathbb{R}^{m \times n}$  are the PAYOFF MATRICES for Alice and Bob.

What should each player do to try and maximise their winnings?

**Example 2.1** (Rock-Scissors-Paper). We have  $S_1 = S_2 = \{\text{Rock, Scissors, Paper}\}$ . Rock beats scissors, scissors beats paper, paper beats rock, and the same choices is a tie. If there is a win, the loser pays the winner ¥1. We may superimpose the payoff matrices as

Alice \ Bob	Bob		
	$\beta_1 = \text{Rock}$	$\beta_2 = \text{Scissors}$	$\beta_3 = \text{Paper}$
$\alpha_1 = \text{Rock}$	(0, 0)	(1, -1)	(-1, 1)
$\alpha_2 = \text{Scissors}$	(-1, 1)	(0, 0)	(1, -1)
$\alpha_3 = \text{Paper}$	(1, -1)	(-1, 1)	(0, 0)

Table 2.1: Rock-Scissor-Paper

(2.1) is also called the PAYOFF MATRICES.

**Example 2.2** (Prisoner's Dilemma). Alice and Bob have been arrested and accused of jointly committing a robbery. The police have offered each of them to either remain silent, or testify the other suspect.

- If they both remain silent, they are both jailed for 1 year.
- If they both testify, they are both jailed for 3 years.

- If one suspect testifies, and the other remains silent, then one who testifies is set free, while the other is jailed for 6 years.

Thus,  $S_1 = S_2 = \{\text{Silent}, \text{Testify}\}$ . They are then taken into separate rooms and questioned, so they are unaware of the other's action.

The payoff matrix is

Alice \ Bob	Bob	
	$\beta_1 = \text{Silent}$	$\beta_2 = \text{Testify}$
$\alpha_1 = \text{Silent}$	$(-1, -1)$	$(-6, 0)$
$\alpha_2 = \text{Testify}$	$(0, -6)$	$(-3, 3)$

Table 2.2: The payoff matrix

In games such as Examples 1 and 2, the following may be assumed:

- The games are full information: Both Alice and Bob know that the other player knows both strategy sets  $S_1, S_2$ , and the payoff matrix.
- The games are “non-cooperative”: Alice and Bob are not in an alliance, and each plays to promote only their own winnings.
- Alice and Bob both play “rationally”. That is, if Alice chooses  $\alpha_i$ , then Bob should try to choose  $\beta_j$  to promote his winnings, and vice versa.

We may also consider dropping any of these assumptions. Such games may be extended to more than two players. These games have applications in economics, political science (eg: voting), computer science, etc.

## 2.2 Two-player zero-sum and constant-sum games

If in the game between Alice and Bob, we have  $A + B = 0$ , then we have a zero-sum game. Thus, Alice wins  $a_{ij} \implies$  Bob loses  $a_{ij}$ , or Bob pays Alice  $a_{ij}$ . Example 1 is a zero-sum game.

It suffices to study only the matrix  $A$ . A “rational” situation may occur as follows. If Alice chooses  $\alpha_i$ , she should expect Bob to choose  $\beta_j$  to maximise his winnings (thus minimise Alice's winnings). So Alice can win  $\geq \max_i \min_j a_{ij}$ . Likewise, if Bob choose  $\beta_j$ , he should expect Alice to choose  $\alpha_i$  to maximise her winnings and Bob's losses. So Bob can lose  $\leq \min_j \max_i a_{ij}$ .

If we have

$$\max_i \min_j a_{ij} = \min_j \max_i a_{ij} = v, \text{ says}$$

then any entry of  $A$  is the same row as  $\max_i \min_j a_{ij}$ , and same column as  $\min_j \max_i a_{ij}$ , is a saddle point, or an equilibrium point. In this case, we say the game has value  $v$ .

Indeed, we have the following result.

**Theorem 2.1.** Suppose we have a zero-sum game between Alice and Bob, where  $A \in \mathbb{R}^{m \times n}$  is Alice's payoff matrix.

- (a) We have  $\max_i \min_j a_{ij} \leq \min_j \max_i a_{ij}$ .
- (b) An entry  $a_{ij} \in A$  is a saddle point if and only if  $a_{ij}$  is the smallest number in row  $i$ , and the largest number in column  $j$ .
- (c) Any two saddle points have the same value.

*Proof.* HW3 □

Thus, if the game has a saddle point with value  $v$ , then we have rational situation that Alice/Bob are guaranteed to win/lose  $v$ , if they choose  $\alpha_i$  and  $\beta_j$  corresponding to a saddle point. If Alice deviates to some  $\alpha_k$ , then Th 2.1 (b)  $\Rightarrow$  her winnings can become  $< v$ . A similar situation applies to Bob if he deviates to some  $\beta_l$ .

**Example 2.3.** Suppose we have the payoff matrix  $A$ :

Alice \ Bob						Row min
		$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	
$\alpha_1$		3	4	10	5	3
$\alpha_2$		8	5	-2	1	-2
$\alpha_3$		6	5	7	9	5 $\leftarrow \max_i \min_j a_{ij}$
Column max $\rightarrow$		8	5 $\leftarrow \min_j \max_i a_{ij}$	10	9	

Table 2.3: Payoff Matrix

We see that there is a saddle point with value 5 as indicated. Note that there are other entries also with value 5. If the game is played rationally, the Alice win 5 units by choosing  $\alpha_3$ , and Bob loses 5 units by choosing  $\beta_2$ .

Similarly, suppose we have the payoff matrices

$$A = \begin{pmatrix} 8 & \boxed{4} & 4 & \boxed{4} \\ 9 & 3 & 2 & \boxed{4} \\ 5 & \boxed{4} & 7 & \boxed{4} \\ 0 & \boxed{4} & 8 & 2 \end{pmatrix} \quad A = \begin{pmatrix} 0 & \boxed{1} & \boxed{-1} \\ \boxed{-1} & 0 & \boxed{1} \\ \boxed{1} & \boxed{-1} & 0 \end{pmatrix}$$

The first matrix has four saddle points with value 4. The second, which is from Example 1, has no saddle point.

Now suppose that  $A + B = C$ , where every entry of  $C$  is the same, say equal to  $c$ . Then we have a constant-sum game. A zero-sum game is thus a constant-sum game with  $c = 0$ . Constant-sum games generalise zero-sum games, since Bob aims to maximise  $b_{ij} = c - a_{ij} \iff$  Bob aims to minimise  $a_{ij}$ . Then notions of rationality, and saddle points of  $A$ , remain valid. We also have Th 2.1 with “constant-sum” in place of “zero-sum”.

Back to a zero-sum game, what happens if the matrix  $A$  has no saddle point? If Alice and Bob still play rationally, then they would use mixed strategy, where they make their choices according to some probability distributions. In the case where they choose  $\alpha_i$  and  $\beta_j$  with probability 1, we have pure strategy. Thus in particular, if  $A$  has a saddle point, then we have pure strategy since Alice and Bob would always choose  $\alpha_i$  and  $\beta_j$  corresponding to a saddle point.

We have

Alice \ Bob	Bob		
	$\beta_1$	$\cdots$	$\beta_n$
$\alpha_1$	$a_{11}$	$\cdots$	$a_{1n}$
$\vdots$	$\vdots$		$\vdots$
$\alpha_m$	$a_{m1}$	$\cdots$	$a_{mn}$

For  $1 \leq i \leq m$ , suppose Alice choose  $\alpha_i$  with probability  $x_i \in [0, 1]$ , so that  $x_1 + \cdots + x_m = 1$ . For example,  $m = 2$ ,  $x_1 = x_2 = \frac{1}{2}$  means Alice throws a fair coin to make her decision. Then Alice’s expected winnings, if Bob chooses  $\beta_j$  ( $1 \leq j \leq n$ ), is  $a_{1j}x_1 + \cdots + a_{mj}x_m$ . So Alice is expected to win at least

$$u = \min\{a_{1j}x_1 + \cdots + a_{mj}x_m : 1 \leq j \leq n\}$$

Thus, Alice wants to choose  $x_1, x_2, \dots, x_m$  s.t.  $u$  is maximised. We have Alice’s LP

$$\begin{aligned} \max \quad & u \\ \text{s.t.} \quad & \begin{cases} a_{11}x_1 + \cdots + a_{m1}x_m \geq u \\ \vdots \\ a_{1n}x_1 + \cdots + a_{mn}x_m \geq u \\ x_1 + \cdots + x_m = 1 \\ x_i \geq 0 \ (1 \leq i \leq m), \quad u \in \mathbb{R} \end{cases} \end{aligned} \tag{2.3}$$

Writing  $u = u_1 - u_2$  where  $u_1, u_2 \geq 0$ , we convert to the form of (1.1):

$$\begin{aligned} \max \quad & c^T \tilde{x} \\ \text{s.t.} \quad & \begin{cases} A\tilde{x} \leq b \\ \tilde{x} \geq 0 \end{cases} \end{aligned} \tag{2.4}$$

where

$$A = \begin{pmatrix} -A^T & \mathbb{1}_n & -\mathbb{1}_n \\ \mathbb{1}_m^T & 0 & 0 \\ -\mathbb{1}_m^T & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+2) \times (m+2)}, \quad b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ -1 \end{pmatrix} \in \mathbb{R}^{n+2}, \quad c = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ -1 \end{pmatrix} \in \mathbb{R}^{m+2}$$

$$\tilde{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \\ u_1 \\ u_2 \end{pmatrix} \in \mathbb{R}^{m+2}, \quad \text{and} \quad \mathbb{1}_k = \underbrace{(1 \ \cdots \ 1)}_k^T$$

Similarly, for  $1 \leq i \leq n$ , suppose Bob choose  $\beta_j$  with probability  $y_j \in [0, 1]$ , so that  $y_1 + \cdots + y_n = 1$ . Then Bob's expected loss, if Alice chooses  $\alpha_i$  ( $1 \leq i \leq m$ ), is  $a_{i1}y_1 + \cdots + a_{in}y_n$ . So Bob is expected to lose at most

$$w = \max\{a_{i1}y_1 + \cdots + a_{in}y_n : 1 \leq i \leq m\}$$

Thus, Bob wants to choose  $y_1, \dots, y_n$  s.t.  $w$  is minimised. We have Bob's LP:

$$\begin{aligned} \min \quad & w \\ \text{s.t.} \quad & \begin{cases} a_{11}y_1 + \cdots + a_{1n}y_n \leq w \\ \vdots \\ a_{m1}y_1 + \cdots + a_{mn}y_n \leq w \\ y_1 + \cdots + y_n = 1 \\ y_j \geq 0 \ (1 \leq j \leq n), \quad w \in \mathbb{R} \end{cases} \end{aligned} \quad (2.5)$$

Writing  $w = w_1 - w_2$ , where  $w_1, w_2 \geq 0$ , we convert to the form of (1.7):

$$\begin{aligned} \min \quad & b^T \tilde{y} \\ \text{s.t.} \quad & \begin{cases} \tilde{A}^T \tilde{y} \succeq c \\ \tilde{y} \succeq 0 \end{cases} \end{aligned} \quad (2.6)$$

where  $\tilde{y} = (y_1 \ \cdots \ y_n \ w_1 \ w_2)^T \in \mathbb{R}^{n+2}$ .

Now clearly (2.4) and (2.6) are dual LPs. It is easy to show that both (2.4) and (2.6) are feasible and bounded. By Th 1.7, we have strong duality. Let both optimal values of (2.4) and (2.6) be  $v$ . We say that  $v$  is the value of the game.

We have the following important result in game theory.

**Theorem 2.2.** (Von Neumann Minimax Theorem) Suppose that Alice and Bob play a zero-sum game rationally, using mixed strategy. Let  $A \in \mathbb{R}^{m \times n}$  be Alice's payoff matrix. Then  $\exists$  optimal probability distributions  $\bar{x} = (x_1, \dots, x_m)$  and  $\bar{y} = (y_1, \dots, y_n)$  for Alice and Bob, s.t. Alice is expected to win  $\geq v$ , and Bob is expected to lose  $\leq v$ , where  $v$  is the value of the game.

Moreover,

$$\max_x \min_y x^T A y = \min_y \max_x x^T A y = v \quad (2.7)$$

where the max and min are taken over  $x \in [0, 1]^m$ ,  $y \in [0, 1]^n$  with  $\sum_{i=1}^m x_i = \sum_{j=1}^n y_j = 1$ . Also,  $\bar{x}$  and  $e_j \in \mathbb{R}^n$  attain  $\max_x \min_y x^T A y = v$  for some  $j$ , and  $e_i \in \mathbb{R}^m$  and  $\bar{y}$  attain  $\min_y \max_x x^T A y = v$  from some

*i.* (**Recall:**  $e_i \in \mathbb{R}^k$  is the standard basis vector,  $e_i = \overbrace{(0 \cdots 1 0 \cdots 0)}^k$ ).

*Proof.* From the above discussion, it suffices to prove (2.7). Note that the set of  $y \in [0, 1]^n$  with  $\sum_{j=1}^n y_j = 1$  defines a simplex in  $\mathbb{R}^n$ , whose extreme points are  $e_1, \dots, e_n$ . For any fixed  $x$ , since  $x^T A y$  is a linear function in  $y$ , we have  $\min_y x^T A y = \min_j x^T A e_j$  by Th 1.2. Thus,

$$\max_x \min_y x^T A y = \max_x \min_j x^T A e_j \stackrel{(2.3), \text{ Th 1.7}}{=} v.$$

Similarly,

$$\min_y \max_x x^T A y = \min_y \max_x x^T A y \stackrel{(2.5), \text{ Th 1.7}}{=} v,$$

and (2.7) holds. □

**Remark 2.1.** • If  $A$  has a saddle point  $a_{ij}$ , then the mixed strategy reduces to pure strategy.

We have the LP (2.3), (2.5) and value  $v$ , where  $\begin{pmatrix} \bar{x} \\ \bar{u} \end{pmatrix} = \begin{pmatrix} e_i \\ v \end{pmatrix} \in \mathbb{R}^{m+1}$ ,  $\begin{pmatrix} \bar{y} \\ \bar{w} \end{pmatrix} = \begin{pmatrix} e_j \\ v \end{pmatrix} \in \mathbb{R}^{n+1}$  are optimal solutions. The two definitions of “value” coincide.

- Since Alice and Bob both know the matrix  $A$ , they also know the optimal probabilities distributions  $\bar{x} \in \mathbb{R}^m$ ,  $\bar{y} \in \mathbb{R}^n$ , since they can be computed from  $A$ .
- If instead, we have a constant-sum game, say  $A + B = C$ , then we may consider the zero-sum game with  $A' + B' = 0$ , where  $A' = A - \frac{1}{2}C$ ,  $B' = B - \frac{1}{2}C$ . We may then apply all arguments above on  $A'$ .

**Example 2.4.** Again, consider (2.1)

Alice \ Bob			
	Rock	Scissors	Paper
Rock	(0, 0)	(1, -1)	(-1, 1)
Scissors	(-1, 1)	(0, 0)	(1, -1)
Paper	(1, -1)	(-1, 1)	(0, 0)

A convenient method to solve for  $\bar{x}, \bar{y}, v$  is as follows:

We add  $|-1| = 1$  to every entry of  $A, B$  to obtain  $\hat{A}, \hat{B}$  ( $-1$  being the most negative entry in  $A$ ). We have a constant-sum game with  $\hat{A}, \hat{B}$ , where  $\hat{A}$  has non-negative entries. We have

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix}.$$

Using a similar argument for (2.3) on  $\hat{A}$ , we have the LP for Alice:

$$\begin{aligned} \max \quad & u' \\ \left\{ \begin{array}{llll} x_1 & & +2x_3 & \geq u' \\ 2x_1 & +x_2 & & \geq u' \\ & 2x_2 & +x_3 & \geq u' \\ x_1 & +x_2 & +x_3 & = 1 \\ x_1, & x_2, & x_3, & u' \geq 0 \end{array} \right. \end{aligned}$$

We note that  $u' \geq 0$  since  $x_1, x_2, x_3$ , and the entries of  $A$  are  $\geq 0$ . Eliminating  $x_3$  gives:

$$\begin{aligned} \max \quad & u' \\ \left\{ \begin{array}{llll} x_1 & +2x_2 & +u' & \leq 2 \\ -2x_1 & +x_2 & +u' & \leq 1 \\ x_2 & -x_2 & +u' & \leq 1 \\ x_1, & x_2, & u' & \geq 0 \end{array} \right. \end{aligned} \tag{2.8}$$

Similarly, using  $\hat{A}$ , similar to (2.5), we have the LP for Bob:

$$\begin{aligned} \min \quad & w' \\ \text{s.t.} \quad & \begin{cases} -y_1 - 2y_2 + w' \geq 0 \\ 2y_1 + y_2 + w' \geq 2 \\ -y_1 + y_2 + w' \geq 1 \\ y_1, y_2, w' \geq 0 \end{cases} \end{aligned} \quad (2.9)$$

We can then use some method to solve (2.8), (2.9), such as the simplex or two-phase methods. We find the optimal solutions  $(\bar{x}, \bar{u}') = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 1)$  and  $(\bar{y}, \bar{w}') = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 1)$ . The value of the original zero-sum game is  $v = \bar{u}' - 1 = \bar{w}' - 1 = 0$ . All of this make sense, since the moves Rock, Scissors and Paper are “symmetric”, so they each should be made with probability  $\frac{1}{3}$ . If the game is played many times, say 100, the overall expected winnings for both players should be zero (the value).

**Definition 2.1** (**DOMINATE**). The strategy  $\alpha_i$  **DOMINATES** the strategy  $\alpha_k$  if  $a_{ij} \geq a_{kj}, \forall 1 \leq j \leq n$ . Similarly,  $\beta_j$  **DOMINATES**  $\beta_l$  if  $b_{ij} \geq b_{il}, \forall 1 \leq i \leq m$ .

We note that in a zero-sum game, if  $\alpha_i$  **DOMINATES**  $\alpha_k$ , then in (2.5), the constraint  $a_{i1}y_1 + \dots + a_{in}y_n \leq w$  is stronger than the constraint  $a_{k1}y_1 + \dots + a_{kn}y_n \leq w$ , and we may ignore the latter. Likewise, if  $\beta_j$  **DOMINATES**  $\beta_l$ , then in (2.3),  $a_{1k}x_1 + \dots + a_{mj}x_m \geq u$  is stronger than  $a_{1l}x_1 + \dots + a_{ml}x_m \geq u$ , and we may ignore the latter.

In other words, to simplify a zero-sum game with matrix  $A$ , we may delete any dominated rows, then any **DOMINATING** columns, and repeat to obtain a simpler matrix.

**Example 2.5.** A zero-sum game has

$$A = \begin{pmatrix} 1 & -3 & -2 \\ -1 & 3 & 2 \\ 4 & -2 & -2 \\ 2 & 1 & 0 \end{pmatrix} \quad (2.10)$$

We see that rows 2 and 3 dominate rows 4 and 1. Delete rows 1 and 4.

$$\begin{pmatrix} -1 & 3 & 2 \\ 4 & -2 & -2 \end{pmatrix}$$



Now, column 2 dominates column 3. Delete column 2.

$$\begin{pmatrix} -1 & 2 \\ 4 & -2 \end{pmatrix}$$

Now proceed as before. Add  $|-2| = 2$  to all entries. Note also the “surviving” variables are  $x_2, x_3, y_1, y_3$ :

$$\begin{pmatrix} 1 & 4 \\ 6 & 0 \end{pmatrix}$$

The LPs in the forms (2.3), (2.5) are

$$\begin{aligned} \max \quad & u' \\ \left\{ \begin{array}{l} x_2 + 6x_3 \geq u' \\ 4x_2 \geq u' \\ x_2 + x_3 = 1 \\ x_2, x_3, u' \geq 0 \end{array} \right. \end{aligned}$$

$$\begin{aligned} \min \quad & w' \\ \left\{ \begin{array}{l} y_1 + 4y_3 \leq w' \\ 6y_1 \leq w' \\ y_1 + y_3 = 1 \\ y_1, y_3, w' \geq 0 \end{array} \right. \end{aligned}$$

Using  $x_3 = 1 - x_2$ ,  $y_3 = 1 - y_2$ , we have

$$\begin{aligned} \max \quad & u' \\ \left\{ \begin{array}{l} 5x_2 + u' \leq 6 \\ -4x_2 + u' \leq 0 \\ x_2, u' \geq 0, \quad x_2 \leq 1 \end{array} \right. \end{aligned}$$

$$\begin{aligned} \min \quad & w' \\ \left\{ \begin{array}{l} 3y_1 + w' \geq 4 \\ -6y_1 + w' \geq 0 \\ y_1, w' \geq 0, \quad y_1 \leq 1 \end{array} \right. \end{aligned}$$

We can easily obtain the optimal solutions  $(\bar{x}_2, \bar{u}') = (\frac{2}{3}, \frac{8}{3})$ ,  $(\bar{y}_1, \bar{w}') = (\frac{4}{9}, \frac{8}{3})$ . For the game (2.10), the optimal probability distributions are  $\bar{x} = (0 \ \frac{2}{3} \ \frac{1}{3} \ 0)^T$ ,  $\bar{y} = (\frac{4}{9} \ 0 \ \frac{5}{9})^T$ , and the value is  $v = \frac{8}{3} - 2 = \frac{2}{3}$ .

## 2.3 Two-player non-constant-sum games

A **NON-CONSTANT-SUM GAME**, often also called a **NON-ZERO-SUM GAME**, is a game which is not constant-sum. Thus Alice and Bob may possibly both win or both lose from a particular pair of strategies.

We continue to assume that Alice and Bob play rationally, and they are non-cooperative.

**Definition 2.2** (Nash Equilibrium). In a two-player game  $(A, B)$ , a **(PURE STRATEGY) NASH EQUILIBRIUM (NE)** is an entry  $(a_{ij}, b_{ij})$  s.t.  $a_{ij} \geq a_{kj}$ ,  $\forall 1 \leq k \leq m$ , and  $b_{ij} \geq b_{il}$ ,  $\forall 1 \leq l \leq n$ . Thus if Alice changes strategy from a NE and Bob remains, then Alice cannot win more, and similarly for Bob. A NE is precisely a saddle point in a constant-sum game.

In the game (2.2),  $(-3, -3)$  is the only NE.

If a game does not have a NE, we may again consider mixed strategy.

**Definition 2.3.** Let

$$\Delta_k = \{z \in [0, 1]^k : \sum_{i=1}^k z_i = 1\}$$

be the **PROBABILITY SIMPLEX**, the space of all probability distributions for  $k$  outcomes.

**Definition 2.4.** Consider the game  $(A, B)$  ( $A, B \in \mathbb{R}^{m \times n}$ ), played with mixed strategy. Let  $x \in \Delta_m, y \in \Delta_n$ .

- For fixed  $x, y$ , the **EXPECT UTILITY** of Alice and Bob are

$$\mathbb{E}_A(x, y) = \sum_{i,j} a_{ij} x_i y_j = x^T A y, \text{ and } \mathbb{E}_B(x, y) = \sum_{i,j} b_{ij} x_i y_j = x^T B y.$$

- $\bar{x} \in \Delta_m$  is Alice's **BEST RESPONSE** to  $y \in \Delta_n$  if

$$\mathbb{E}_A(\bar{x}, y) \geq \mathbb{E}_B(x, y), \forall x \in \Delta_m.$$

Similarly for  $\bar{y} \in \Delta_n$  for Bob:  $\mathbb{E}_B(x, \bar{y}) \geq \mathbb{E}_B(x, y), \forall y \in \Delta_n$ .

- $(\bar{x}, \bar{y})$  is a **MIXED STRATEGY NASH EQUILIBRIUM (MSNE)** if  $\bar{x}$  is Alice's best response to  $\bar{y}$ , and  $\bar{y}$  is Bob's best response to  $\bar{x}$ . The **NASH EQUILIBRIUM PAYOFFS** for Alice and Bob are  $\mathbb{E}_A(\bar{x}, \bar{y})$  and  $\mathbb{E}_B(\bar{x}, \bar{y})$ .

**Example 2.6.** Consider

Alice \ Bob	$\beta_1$	$\beta_2$
	$\alpha_1$	$\alpha_2$
$\alpha_1$	$(2, -3)$	$(1, 2)$
$\alpha_2$	$(1, 1)$	$(4, -1)$

Suppose Alice and Bob choose  $\alpha_1, \beta_1$  with probabilities  $p, q \in [0, 1]$ . Then

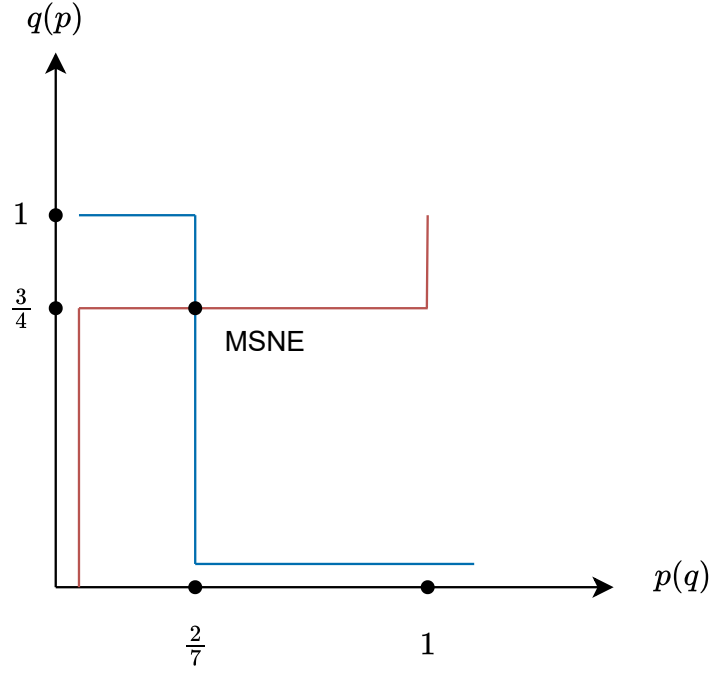
$$\begin{aligned}\mathbb{E}_A(p, q) &= 2pq + p(1 - q) + (1 - p)q + 4(1 - p)(1 - q) \\ &= (4q - 3)p + (-3q + 4)\end{aligned}$$

$$\implies \text{Best response } p = \begin{cases} 0, & \text{if } q < \frac{3}{4} \\ 1, & \text{if } q > \frac{3}{4} \\ \in [0, 1] & \text{if } q = \frac{3}{4} \end{cases}$$

$$\begin{aligned}\mathbb{E}_B(p, q) &= -3pq + 2p(1 - q) + (1 - p)q - (1 - p)(1 - q) \\ &= (-7p + 2)q + (3p - 1)\end{aligned}$$

$$\implies \text{Best response } q = \begin{cases} 0, & \text{if } p > \frac{2}{7} \\ 1, & \text{if } p < \frac{2}{7} \\ \in [0, 1], & \text{if } p = \frac{2}{7} \end{cases}$$

We have



and one MSNE  $\bar{x} = (\frac{2}{7}, \frac{5}{7})$ ,  $\bar{y} = (\frac{3}{4}, \frac{1}{4})$ . The NE payoffs for Alice and Bob are  $\mathbb{E}_A(\frac{2}{7}, \frac{3}{4}) = \frac{7}{4}$ ,  $\mathbb{E}_B(\frac{2}{7}, \frac{3}{4}) = -\frac{1}{7}$ .

**Definition 2.5.** The strategies  $\alpha_{i_1}, \dots, \alpha_{i_t}$  STRICTLY DOMINATE the strategy  $\alpha_k$  ( $k \neq i_1, i_2, \dots, i_t$ ) if  $\exists \gamma \in \Delta_t$ , s.t.

$$\gamma_1 a_{i_1, j} + \dots + \gamma_t a_{i_t, j} > a_{kj}, \quad \forall 1 \leq j \leq n$$

We similar define  $\beta_{j_1}, \dots, \beta_{j_t}$  STRICTLY DOMINATE  $\beta_l$ .

**Lemma 2.3.** Suppose that in the game  $(A, B)$ , the strategies  $\alpha_{i_1}, \dots, \alpha_{i_t}$  strictly dominate the strategy  $\alpha_k$ . Then every MSNE  $(\bar{x}, \bar{y})$  must have  $\bar{x}_k = 0$ . Similarly, if  $\beta_{j_1}, \dots, \beta_{j_t}$  strictly dominates  $\beta_l$ , then every MSNE  $(\bar{x}, \bar{y})$  must have  $\bar{y}_l = 0$ .

*Proof.* By permuting the  $\alpha_i$ , we may assume  $\alpha_1, \dots, \alpha_t$  ( $1 \leq t < m$ ) strictly dominate  $\alpha_m$ . Then  $\exists \gamma \in \Delta_t$ , s.t.  $\gamma_1 a_{1j} + \dots + \gamma_t a_{tj} > a_{mj}$ ,  $\forall 1 \leq j \leq n$ . Suppose  $\exists$  a MSNE  $(\bar{x}, \bar{y})$  s.t.  $\exists \bar{x}_m > 0$ . Consider  $x' \in \Delta_m$  where

$$x'_i = \begin{cases} x_1 + \varepsilon \gamma_i \bar{x}_m, & \text{if } 1 \leq i \leq t, \\ \bar{x}_i & \text{if } t < i < m, \\ (1 - \varepsilon) \bar{x}_m & \text{if } i = m \end{cases}$$

for some small  $\varepsilon > 0$ . Note that  $x' \in \Delta_m$  since  $\bar{x}_m > 0 \implies \bar{x}_i < 1, \forall 1 \leq i \leq m$ . Then

$$\begin{aligned} \mathbb{E}_A(x', \bar{y}) &= \sum_{\substack{1 \leq i \leq t, \\ 1 \leq j \leq n}} a_{ij}(\bar{x}_i + \varepsilon \gamma_i \bar{x}_m) \bar{y}_j + \sum_{\substack{t < i < m, \\ 1 \leq j \leq n}} a_{ij} \bar{x}_i \bar{y}_j + \sum_{1 \leq j \leq n} a_{mj}(1 - \varepsilon) \bar{x}_m \bar{y}_j \\ &= \sum_{\substack{1 \leq i < m, \\ 1 \leq j \leq n}} a_{ij} \bar{x}_i \bar{y}_j + \sum_{1 \leq j \leq n} a_{mj}(1 - \varepsilon) \bar{x}_m \bar{y}_j + \sum_{\substack{1 \leq i \leq t, \\ 1 \leq j \leq n}} a_{ij} \varepsilon \gamma_i \bar{x}_m \bar{y}_j \\ &\stackrel{(*)}{>} \sum_{\substack{1 \leq i < m, \\ 1 \leq j \leq n}} a_{ij} \bar{x}_i \bar{y}_j + \sum_{1 \leq j \leq n} a_{mj}(1 - \varepsilon) \bar{x}_m \bar{y}_j + \sum_{1 \leq j \leq n} a_{mj} \varepsilon \bar{x}_m \bar{y}_j = \mathbb{E}_A(\bar{x}, \bar{y}) \end{aligned}$$

which contradicts that  $\bar{x}$  is Alice's best response to  $\bar{y}$ .

Note that  $(*)$  holds since  $\bar{x}_m \bar{y}_j > 0$  for some  $j$ . A similar proof holds if there is a strict domination among the  $\beta_j$ .  $\square$

**Example 2.6.** Find the MSNEs of

$$\begin{pmatrix} (-1, -1) & (1, -2) & (5, 1) \\ (1, 2) & (2, 1) & (2, 4) \\ (0, -2) & (1, -3) & (3, 2) \\ (3, -2) & (4, 0) & (2, -2) \end{pmatrix}$$

We say that  $(\frac{1}{3} \times \text{row 1}) + (\frac{2}{3} \times \text{row 4})$  and  $(\frac{1}{2} \times \text{row 1}) + (\frac{1}{2} \times \text{row 4})$  strictly dominate rows 2 and 3. Delete row 2 and 3:

$$\begin{pmatrix} (-1, -1) & (1, -2) & (5, 1) \\ (3, -2) & (4, 0) & (2, -2) \end{pmatrix}$$

Next,  $(\frac{1}{2} \times \text{column 2}) + (\frac{1}{2} \times \text{column 3})$  strictly dominates column 1. Delete column 1.

	$y_2$	$y_3$
$x_1$	(1, -2)	(5, 1)
$x_4$	(4, 0)	(2, -2)

The surviving variables are  $x_1, x_4, y_2, y_3$ . Now proceed as in Example (2.6). Suppose Alice and Bob choose  $\alpha_1, \beta_2$  with probabilities  $p, q \in [0, 1]$ . Then

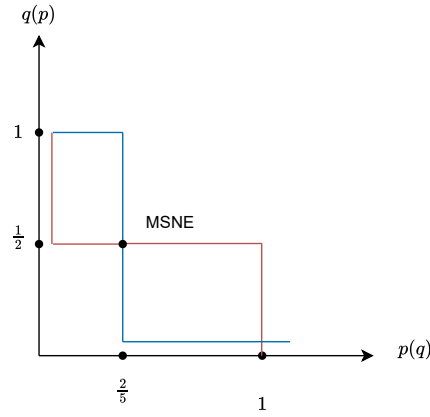
$$\begin{aligned} \mathbb{E}_A(p, q) &= pq + 5p(1 - q) + 4(1 - p)q + 2(1 - p)(1 - q) \\ &= (-6q + 3)p + (2q + 2) \end{aligned}$$

$$\Rightarrow \text{Best response } p = \begin{cases} 0, & \text{if } q > \frac{1}{2}, \\ 1, & \text{if } q < \frac{1}{2}, \\ \in [0, 1], & \text{if } q = \frac{1}{2}. \end{cases}$$

$$\begin{aligned} \mathbb{E}_B(p, q) &= -2pq + p(1 - q) - 2(1 - p)(1 - q) \\ &= (-5p + 2)q + (3p - 2) \end{aligned}$$

$$\Rightarrow \text{Best response } q = \begin{cases} 0, & \text{if } p > \frac{2}{5}, \\ 1, & \text{if } p < \frac{2}{5}, \\ \in [0, 1], & \text{if } p = \frac{2}{5}. \end{cases}$$

We have



There are three MSNEs, two of which are also NEs.

MSNE	NE	NE payoffs	
		Alice	Bob
$\bar{x} = (0 \ 0 \ 0 \ 1)^T, \bar{y} = (0 \ 1 \ 0)^T$	(4, 0)	$\mathbb{E}_A(0, 1) = 4$	$\mathbb{E}_B(0, 1) = 0$
$\bar{x} = (1 \ 0 \ 0 \ 0)^T, \bar{y} = (0 \ 0 \ 1)^T$	(5, 1)	$\mathbb{E}_A(1, 0) = 5$	$\mathbb{E}_B(0, 1) = 1$
$\bar{x} = (\frac{2}{5} \ 0 \ 0 \ \frac{3}{5})^T, \bar{y} = (0 \ \frac{1}{2} \ \frac{1}{2})^T$	\	$\mathbb{E}_A(\frac{2}{5}, \frac{1}{2}) = 3$	$\mathbb{E}_B(\frac{2}{5}, \frac{1}{2}) = -\frac{4}{5}$

### 2.3.1 Cooperative Games

For non-constant-sum games, it is interesting if the players can “cooperate”.

**Example 2.7.** Recall that in Prisoner’s Dilemma, we have

Alice \ Bob		
	$\beta_1 = \text{Silent}$	$\beta_2 = \text{Testify}$
$\alpha_1 = \text{Silent}$	$(-1, -1)$	$(-6, 0)$
$\alpha_2 = \text{Testify}$	$(0, -6)$	$(-3, 3)$

Table 2.4: (2.2)

Suppose that, before being taken into separate rooms and questioned, Alice and Bob can communicate privately with each other. From the table (2.2), it seems a good compromise for them to both remain silence, so they each spend a short time of 1 year in jail.

**BUT**, can they keep their promise (cooperate) when they are questioned separately? If one of them decides to defect and testify (non-cooperate), while the other keeps the promise, the one who defects “gains” and does not go to jail, while the other is “suckered”, and gets 6 years in jail. However, if they defect, then they are both worse off than if they had cooperated, since they each have to go to jail for 3 years ( $3 > 1$ ).

We see that Prisoner’s Dilemma is interesting since it asks the question “Can Alice and Bob really trust each other?”.

In general, the payoff matrix may be represented as follows:

Alice \ Bob	C	NC
C	$(r, r)$	$(s, t)$
NC	$(t, s)$	$(p, p)$

C = Cooperate

NC = Non-Cooperate

r = Reward for cooperating

t = Temptation for defecting

s = Punishment for getting suckered

p = Punishment for non-cooperation

We see that  $t$  and  $s$  should be the largest and smallest payoffs. Also, for cooperation to be possible, we need  $r > p$ . Thus,  $t > r > p > s$ . We see that  $(p, p)$  is the only NE. Moreover, by strict domination, we see that  $\bar{x} = \bar{y} = (0 \ 1)^T$  is the only MSNE, giving the NE payoffs  $p, p$ . Thus, it is wisest if they both defect.

**Example 2.7** (The game of Chicken). After a tough F1 race, Livid Lewis accuses Mad Max of cheating towards to win. Lewis promises Max that in the next race, he will cause a collision so that Max cannot win the race, he will cause a collision so that Max cannot win the race, to which Max replies that he will also promise to cause a collision.

In the next race, the pair are side by side in their cars.

- In one of them swerves away<sup>a</sup> to avoid a collision (“chickens<sup>b</sup> out”), and the other remains committed to the promise, the one who swerves is overtaken by the other. The score “mind games points” of  $-3$  and  $5$ .
- If they both swerve, they both spin out of the race, and they each score  $0$  mind games points.
- But is they both kept the promise, they cause a collision and each suffers an injury. They both score  $-10$  mind games points.

The payoff matrix is

Lewis \ Max	Don't swerve	Swerve
	$(-10, -10)$	$(5, -3)$
Don't swerve	$(-10, -10)$	$(5, -3)$
Swerve	$(-3, 5)$	$(0, 0)$

We may generally represent the matrix as:

Lewis \ Max	C	NC
	$(r, r)$	$(s, t)$
C	$(r, r)$	$(s, t)$
NC	$(t, s)$	$(p, p)$

Clearly,  $s$  and  $r$  should be the largest and smallest payoffs. We should have  $p > t$ , since if one chickens out, it is better if the other chickens out than not. Thus  $s > p > t > r$ . We see that  $(t, s)$  and  $(s, t)$  are NEs. If say  $s = 2$ ,  $p = 1$ ,  $t = -1$ ,  $r = -M$  ( $M > 0$  is large), then we can find that  $\bar{x} = \bar{y} = (\frac{1}{M}, 1 - \frac{1}{M})$  is a MSNE. Thus it is wisest if they both swerve.

<sup>a</sup>突然转向

<sup>b</sup>a man who is not brave

## 2.4 Games with $t$ players

Suppose  $t \geq 2$  players play a game. Their strategy sets are  $S_1, \dots, S_t$ . Let  $S = S_1 \times \dots \times S_t$ . The **PAYOFF FUNCTION** for Player  $i$  ( $1 \leq i \leq t$ ) is  $u_i : S \rightarrow \mathbb{R}$ , where Player  $i$  wins  $u_i(s)$  when the strategies



$s = (s_1, \dots, s_t) \in S$  are chosen by the  $t$  players. Thus for a 2-player game  $(A, B)$ ,  $u_1, u_2$  correspond to  $A, B$ .

We assume the players are non-cooperative, and they play rationally. Also, the sets  $S_i$  are finite.

**Definition 2.6.** • For  $s \in S$  ( $1 \leq i \leq t$ ), write

$$s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_t) \in S_1 \times \dots \times S_{i-1} \times S_i \times \dots \times S_t$$

For  $s'_i \in S_i$ , write

$$(s'_i, s_{-i}) = (s_1, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_t) \in S$$

- A (pure strategy) Nash equilibrium (NE) is  $\bar{s} \in S$  (or  $(u_1(\bar{s}), \dots, u_t(\bar{s}))$ ) s.t.

$$u_i(\bar{s}) = u_i(\bar{s}_i, \bar{s}_{-i}) \geq u_i(s_i, \bar{s}_{-i}), \quad \forall 1 \leq i \leq t.$$

Thus if Player  $i$  changes strategy from a NE and all other players remain, then Player  $i$  cannot win more.

**Definition 2.7.** • For probability distributions  $\pi_i : S_i \rightarrow [0, 1]$  ( $1 \leq i \leq t$ ), the EXPECTED UTILITY of Player  $i$  is

$$\mathbb{E}_i(\pi_1, \dots, \pi_t) = \sum_{s \in S} u_i(s) \prod_{j=1}^t \pi_j(s_j).$$

- $\bar{p}_i$  is Player  $i$ 's best response to  $\pi_1, \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_t$  if

$$\mathbb{E}_i(\pi_1, \dots, \pi_{i-1}, \bar{p}_i, \pi_{i+1}, \dots, \pi_t) \geq \mathbb{E}_i(\pi_1, \dots, \pi_{i-1}, \pi_i, \pi_{i+1}, \dots, \pi_t), \quad \forall \pi_i$$

- $(\bar{\pi}_1, \dots, \bar{\pi}_t)$  is a MIXED STRATEGY NASH EQUILIBRIUM (MSNE) if  $\bar{\pi}_i$  is Player  $i$ 's best response to  $\bar{\pi}_1, \dots, \bar{\pi}_{i-1}, \bar{\pi}_{i+1}, \dots, \bar{\pi}_t$ ,  $\forall 1 \leq i \leq t$ . The NASH EQUILIBRIUM PAYOFF for Player  $i$  is  $\mathbb{E}_i(\bar{\pi}_1, \dots, \bar{\pi}_t)$ .

**Example 2.8.** Suppose we have a 3-player game, where  $S_1 = \{\alpha_1, \alpha_2\}$ ,  $S_2 = \{\beta_1, \beta_2\}$ ,  $S_3 = \{\gamma_1, \gamma_2\}$ , and

$P_1 \backslash P_2$	$\beta_1$	$\beta_2$
$\alpha_1$	(3, 3, 3)	(3, 4, 3)
$\alpha_2$	(4, 3, 3)	(1, 1, 0)

Table 2.5: If  $P_3$  choose  $\gamma_1$ 

$P_1 \backslash P_2$	$\beta_1$	$\beta_2$
$\alpha_1$	(3, 3, 4)	(0, 1, 1)
$\alpha_2$	(1, 0, 1)	(1, 1, 1)

Table 2.6: If  $P_3$  choose  $\gamma_3$ 

There are four NEs as indicated. Now consider mixed strategies. Let  $x = (p, 1-p)$ ,  $y = (q, 1-q)$ ,  $z = (r, 1-r)$ , be the probability distributions.

$$\begin{aligned}
 \mathbb{E}_1(p, q, r) &= [3qr + 3(1-q)r + 3q(1-r)p]p \\
 &\quad + [4qr + (1-q)r + q(1-r) + (1-q)(1-r)](1-p) \\
 &= (3qr + 3r - 6qr - 1)p + (3qr + 1)
 \end{aligned}$$

$$\Rightarrow \text{Best response } p = \begin{cases} 0, & \text{if } 3qr + 3r - 6qr - 1 < 0 (*) \\ 1, & \text{if } 3qr + 3r - 6qr - 1 > 0 \\ \in [0, 1], & \text{if } 3qr + 3r - 6qr - 1 = 0 \end{cases}$$

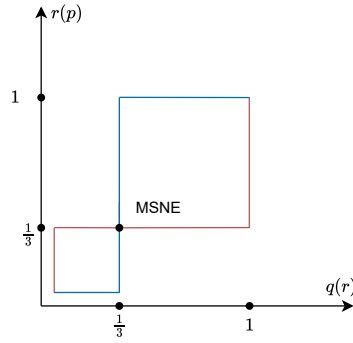
Similarly (by symmetry of  $p, q, r$ )

$$\begin{aligned}
 \Rightarrow \text{BR } q &= \begin{cases} 0, & \text{if } 3p + 3r - 6pr - 1 < 0 \\ 1, & \text{if } 3p + 3r - 6pr - 1 > 0 \\ \in [0, 1] & \text{if } 3p + 3r - 6pr - 1 = 0 \end{cases} \\
 \Rightarrow \text{BR } r &= \begin{cases} 0, & \text{if } 3p + 3q - 6pq - 1 < 0 \\ 1, & \text{if } 3p + 3q - 6pq - 1 > 0 \\ \in [0, 1] & \text{if } 3p + 3q - 6pq - 1 = 0 \end{cases}
 \end{aligned}$$

Let  $(\bar{x}, \bar{y}, \bar{z})$  be a MSNE. If  $\bar{p} = 0$ ,

$$\text{BR } q = \begin{cases} 0, & \text{if } r < \frac{1}{3} \\ 1, & \text{if } r > \frac{1}{3} \\ \in [0, 1], & \text{if } r = \frac{1}{3} \end{cases},$$

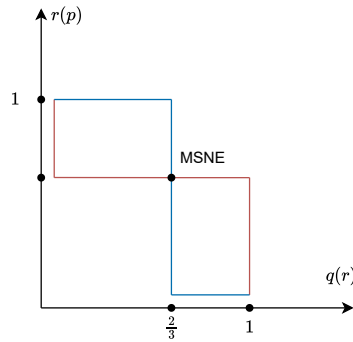
$$\text{BR } e = \begin{cases} 0, & \text{if } q < \frac{1}{3} \\ 1, & \text{if } q > \frac{1}{3} \\ \in [0, 1], & \text{if } q = \frac{1}{3} \end{cases}.$$



$\Rightarrow (p, q, r) = (0, 0, 0), (0, \frac{1}{3}, \frac{1}{3}), (0, 1, 1)$ . We discard  $(0, \frac{1}{3}, \frac{1}{3})$  since it does not satisfy (\*). If  $\bar{p} = 1$ ,

$$\text{BR } q = \begin{cases} 0, & \text{if } r > \frac{2}{3} \\ 1, & \text{if } r < \frac{2}{3} \\ \in [0, 1], & \text{if } r = \frac{2}{3} \end{cases},$$

$$\text{BR } r = \begin{cases} 0, & \text{if } q > \frac{2}{3} \\ 1, & \text{if } q < \frac{2}{3} \\ \in [0, 1], & \text{if } q = \frac{2}{3} \end{cases}$$



$\Rightarrow (\bar{p}, \bar{q}, \bar{r}) = (1, 0, 1), (1, \frac{2}{3}, \frac{2}{3}), (1, 1, 0)$ .

Now by symmetry of  $p, q, r$ , suppose  $\bar{p}, \bar{q}, \bar{r} \in (0, 1)$ . Then

$$3q + 3r - 6qr - 1 = 0 \quad (1)$$

$$3p + 3r - 6pr - 1 = 0 \quad (2)$$

$$3p + 3q - 6pq - 1 = 0 \quad (3)$$

$$(1), (2) \implies 3q - 6qr = 3p - 6pr \implies (p - q)(1 - 2r) = 0.$$

$$\bullet \text{ If } p = q, (3) \implies 6p - 6p^2 - 1 = 0 \implies \bar{p} = \bar{q} = \frac{3 \pm \sqrt{3}}{6} \in (0, 1).$$

$$(1) \implies \bar{p} = \bar{q} = \bar{r} = \frac{3 \pm \sqrt{3}}{6} \in (0, 1).$$

$$\bullet \text{ If } r = \frac{1}{2}, (1) \implies \frac{1}{2} = 0, \text{ which is absurd.}$$

**Conclusion** NEs:  $(\bar{p}, \bar{q}, \bar{r}) = (0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)$ .

MSNEs:  $(\bar{p}, \bar{q}, \bar{r}) = (1, \frac{2}{3}, \frac{2}{3}), (\frac{2}{3}, 1, \frac{2}{3}), (\frac{2}{3}, \frac{2}{3}, 1), \left(\frac{3 \pm \sqrt{3}}{6}, \frac{3 \pm \sqrt{3}}{6}, \frac{3 \pm \sqrt{3}}{6}\right)$ .

We may obtain all possible  $(\bar{x}, \bar{y}, \bar{z})$  and NE payoffs from these.

We have the celebrated result.

**Theorem 2.4** (Nash's theorem). If a game with  $t \geq 2$  players has finite strategy sets  $S_1, \dots, S_t$  then  $\exists$  MSNE.

Note that 2.4 says that a MSNE exists, but does not tell us how to find one!

A proof of Th 2.4 requires:

**Theorem 2.5** (Brouwer's fixed-point theorem). Let  $K \subset \mathbb{R}^n$  be a compact, convex set, and  $f : K \rightarrow K$  be continuous. Then  $\exists x \in K$ , s.t.  $f(x) = x$ . Such an  $x$  is a **FIXED POINT**.

*Proof.* For Th 2.4 (Using Th 2.5)

First, consider the case of two players Alice and Bob, with matrices  $A, B \in \mathbb{R}^{m \times n}$ . Let  $K = \Delta_m \times \Delta_n$ . For  $(x, y) \in K$ , we define  $f(x, y) = (x', y')$  as follows. Let

$$c_i = c_i(x, y) = \max(A_i y - x^T A y, 0), \quad 1 \leq i \leq m$$

where  $A_i$  is the  $i^{\text{th}}$  row of  $A$ . Note that  $c_i$  is Alice's possible gain by switching from mixed strategy  $x$  to pure strategy  $\alpha_i$ , if such a gain is positive. Let  $x' \in \Delta_m$  where

$$x'_i = \frac{x_i + c_i}{1 + \sum_{k=1}^m c_k}$$

Similarly, let

$$d_j = d_j(x, y) = \max(x^T B^j - x^T B y, 0), \quad 1 \leq j \leq n$$

where  $B^j$  is the  $j^{\text{th}}$  column of  $B$ , and let  $y' \in \Delta_n$ , where

$$y'_j = \frac{y_j + d_j}{1 + \sum_{k=1}^n d_k}.$$

Now, if  $c_i = 0$  (or  $x^T A y \geq A_i y$ ),  $\forall i$ , then  $x' = x$  is a best response to  $y$ . Otherwise,  $c = \sum_{i=1}^m c_i > 0$ . Then

$$\begin{aligned} \sum_{i=1}^m c_i A y &> c x^T A y = \sum_{i=1}^m c_i x^T A y \\ \implies \sum_{i=1}^m (x_i + c_i) y &> (1 + c) x^T A y \\ \implies \sum_{i=1}^m x'_i A_i y &> x^T A y \end{aligned}$$

so  $x'$  is a better response to  $y$  than  $x$  is. Similarly, either  $y' = y$  is a best response to  $x$ , or  $y'$  is a better response to  $x$  than  $y$  is.

Clearly,  $K$  is compact and convex, and  $f$  is continuous since  $c_i$  and  $d_j$  are. Thus Th 2.5  $\implies$   $f$  has a fixed point, which is a MSNE.

For  $t \geq 3$  players, we define for Player  $j$  with pure strategy  $l$ , the quantity  $c_l^{(j)}$ , which is the gain that Player  $j$  gets by switching from current strategy  $x^{(j)}$  to pure strategy  $l$ , is positive, while fixing the current strategies of all other players. The argument is similar.  $\square$



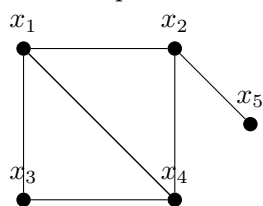
# Chapter 3

## Graph Theory and Networks

### 3.1 Terminology

A GRAPH is a pair  $G = (V, E)$  where  $V$  is a set, and  $E$  is a set of ordered pairs from  $V$ .

For example



$$V = \{x_1, x_2, x_3, x_4, x_5\}$$

$$E = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_1\}, \{x_1, x_5\}, \{x_2, x_5\}, \{x_3, x_5\}, \{x_4, x_5\}\}$$

$V$  is the VERTEX SET and  $E$  is the EDGE SET. We may also write  $V(G)$  and  $E(G)$  for  $V$  and  $E$ .  $|V|$  and  $|E|$  are the ORDER and SIZE of  $G$ .

Note: No loops.

No multi edges.

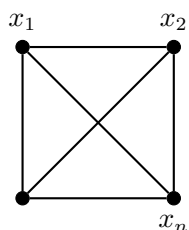
**Important examples:** The EMPTY GRAPH  $E_n$  of order  $n$ :



$$V = \{x_1, \dots, x_n\}$$



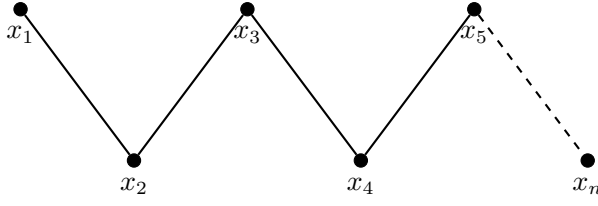
The COMPLETE GRAPH  $K_n$  of order  $n$ :



$$V = \{x_1, \dots, x_n\}, \quad E = \{x_i x_j \mid 1 \leq i < j \leq n\}$$

$$|V| = n, \quad |E| = \binom{n}{2}$$

The PATH  $P_n$  of order  $n$ , LENGTH  $n - 1$ :

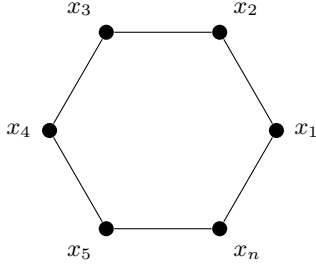


$$V = \{x_1, \dots, x_n\}, \quad E = \{x_i x_{i+1} \mid 1 \leq i \leq n-1\}$$

$$|V| = n, \quad |E| = n - 1$$

We may write  $P_n = x_1 x_2 \cdots x_n$ . We say that  $x_1, x_n$  are the END-VERTICES of  $P_n$ , and  $P_n$  is an  $x_1 - x_n$  PATH.

The CYCLE  $C_n$  of order and length  $n \geq 3$ :

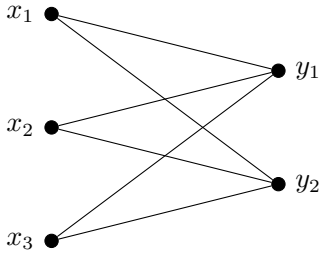


$$V = \{x_1, \dots, x_n\}, \quad E = \{x_i x_{i+1} \mid 1 \leq i \leq n\} \cup \{x_n x_1\}$$

$$|V| = n, \quad |E| = n$$

We may write  $C_n = x_1 x_2 \cdots x_n x_1$ .

The COMPLETE BIPARTITE GRAPH  $K_{m,n}$ :



$$V = X \cup Y, \text{ where } X = \{x_1, \dots, x_m\}, \quad Y = \{y_1, \dots, y_n\}$$

$$E = \{x_i y_j \mid 1 \leq i \leq m, 1 \leq j \leq n\}$$

$$|V| = m + n, \quad |E| = mn$$

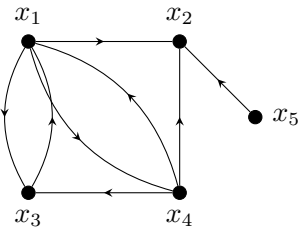
In general, a graph  $G = (V, E)$  is BIPARTITE if we have  $V = X \cup Y$  s.t.  $E \subset \{xy \mid x \in X, y \in Y\}$ .

$G' = (V', E')$  is a SUBGRAPH of  $G = (V, E)$ , written  $G' \subset G$ , if  $V' \subset V$  and  $E' \subset E$ . If  $V' = V$  and  $E' \subset E$ , then  $G'$  is a SPANNING SUBGRAPH of  $G$ .

Let  $G = (V, E)$ . If  $xy \in E$ , then  $x$  and  $y$  are NEIGHBOURS or ADJACENT. The NEIGHBOURHOOD of  $x \in V$  is  $\Gamma(x)$  (or  $N(x) = \{y \in V \mid xy \in E\}$ ). The DEGREE of  $x$  is  $d(x) = |\Gamma(x)|$ .

**Definition 3.1** (Directed Graph). A DIRECTED GRAPH or DIGRAPH is a pair  $D = (V, A)$ , where  $V$  is a set and  $A$  is a set of ordered pairs from  $V$ .

For example:



$$V = \{x_1, \dots, x_5\},$$

$$E = \{x_1 x_2, x_1 x_3, x_1 x_4, x_3 x_1, x_4 x_1, x_4 x_2, x_4 x_3, x_5 x_2\}$$

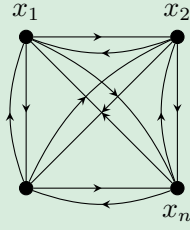
$V$  is the VERTEX SET, and  $A$  is the ARC SET. We may also write  $V(D)$  and  $A(D)$  for  $V$  and  $A$ .  $|V|$  and  $|A|$  are the ORDER and SIZE of  $D$ .



Note: No loops.

No multiple arcs. But  $x_1 - x_2, x_2 - x_1$  is acceptable.

**Example 3.1.** The COMPLETE DIAGRAPH  $K_n$  of order  $n$ :

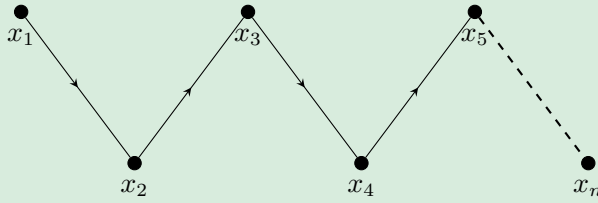


$$V = \{x_1, \dots, x_n\},$$

$$E = \{x_i x_j \mid 1 \leq i \neq j \leq n\}$$

$$|V| = n, |A| = n(n-1)$$

**Example 3.2.** The DIRECTED PATH, or PATH  $\hat{P}_n$  of order  $n$ , length  $n - 1$ :



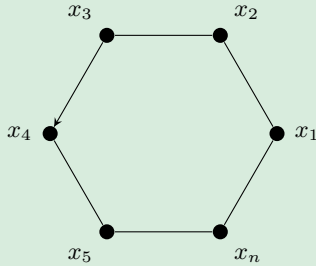
$$V = \{x_1, \dots, x_n\},$$

$$E = \{x_i x_{i+1} \mid 1 \leq i \leq n-1\},$$

$$|V| = n, |E| = n - 1$$

We may write  $\hat{P}_n = x_1 x_2 \cdots x_n$ . We say that  $x_1, x_n$  are the SOURCE and SINK of  $\hat{P}_n$ , and  $\hat{P}_n$  is an  $x_1 - x_n$  Path.

**Example 3.3.** The DIRECTED CYCLE, or DICYCLE  $\hat{C}_n$  of order and length  $n \geq 3$  (Sometimes  $n = 2$  are allowed).



$$V = \{x_1, \dots, x_n\}, E = \{x_i x_{i+1} \mid 1 \leq i \leq n\} \cup \{x_n x_1\}$$

$$|V| = n, |E| = n$$

We may write  $\hat{C}_n = x_1 x_2 \cdots x_n x_1$ .

$D' = (V', A')$  is a SUBDIGRAPH of  $D = (V, A)$ , written  $D' \subset D$ , if  $V' \subset V$  and  $A' \subset A$ . If  $V' = V$  and  $A' \subset A$ , then  $D'$  is a SPANNING SUBDIGRAPH of  $D$ .

Let  $D = (V, A)$ . If  $xy \in A$ , then  $y$  is an OUT-NEIGHBOUR of  $x$  and  $x$  is an IN-NEIGHBOUR of  $y$ . The OUT-NEIGHBOURHOOD/IN-NEIGHBOURHOOD of  $x \in V$  are  $\Gamma^+(x) = \{y \in V \mid xy \in A\}$  and  $\Gamma^-(x) = \{y \in V \mid yx \in A\}$ . The OUT-DEGREE/IN-DEGREE of  $x \in V$  are  $d^+(x) = |\Gamma^+(x)|$  and  $d^-(x) = |\Gamma^-(x)|$ .

For a digraph  $D = (V, A)$ , the UNDERLYING GRAPH is  $G = (V, E)$  where  $E = \{xy \mid xy \in A \text{ or } yx \in A\}$ .

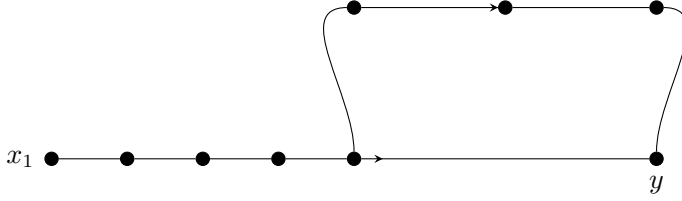
A graph  $G = (V, E)$  is CONNECTED if  $\forall x, y \in V, \exists x - y$  path  $P \subset G$ . A digraph  $D = (V, A)$  is CONNECTED if the underlying graph of  $D$  is connected, and STRONG-CONNECTED if  $\forall x, y \in V, \exists x - y$  path  $\hat{P} \subset D$ .

A **NETWORK GRAPH** is a pair  $(G, w)$ , where  $G$  is a graph, and  $w : E(G) \rightarrow \mathbb{R}$ . A **NETWORK DIGRAPH** is a pair  $(D, w)$ , where  $D$  is a digraph, and  $w : A(D) \rightarrow \mathbb{R}$ . In each case,  $w$  is a **WEIGHT FUNCTION**.

From now on, we assume  $V$  is finite,  $\forall$  graphs and digraphs. In many situations, we consider connected graphs and digraphs.

## 3.2 Shortest path problem

**Definition 3.2.** Let  $(D, w)$  be a network digraph and  $x, y \in V(D)$ . A **WALK** from  $x$  to  $y$ , or an  **$x - y$  WALK**, is a sequence vertices  $x = v_0, v_1, \dots, v_k = y$  s.t.  $v_{i-1}v_i \in A(D)$ ,  $\forall 1 \leq i \leq k$ . Note that  $v_0, v_1, \dots, v_k$  and  $v_0v_1 \dots v_{k-1}v_k$  are not necessarily distinct.



We say that  $y$  is **REACHABLE** from  $x$  if  $\exists$  an  $x - y$  walk (equivalently,  $\exists x - y$  path) in  $D$ . In this case, a **SHORTEST PATH** from  $x$  to  $y$  is an  $x - y$  walk.  $x = v_0, v_1, \dots, v_k = y$  with minimum weight, that is,  $\sum_{i=1}^k w(v_{i-1}, v_i)$  is minimum. Note that  $D$  is strong connected  $\iff y$  is reachable from  $x$ ,  $\forall x, y \in V(D)$ .

We consider the following problem.

### 3.2.1 Problem 3.1 Shortest path problem

Let  $(D, w)$  be a network digraph. Let  $x \in V(D)$  s.t.  $\forall y \in V(D) \setminus \{x\}$ ,  $y$  is reachable from  $x$ . Find a shortest path from  $x$  to every  $y \in V(D) \setminus \{x\}$ . We call  $x$  the **SOURCE VERTEX**.

We consider two algorithms for solving Problem 3.1.

#### A. Dijkstra's Algorithm

We assume  $w : A(D) \rightarrow \mathbb{R}_{\geq 0}$  in Problem 3.1. As Dijkstra's Algorithm is performed, every vertex  $y \in V(D)$  is given a temporary label  $T(y)$  which gets updated, then eventually a permanent label  $P(y)$ , which will be the weight of a shortest path from  $x$  to  $y$ . Then algorithm is as follows.

1. Set  $P(x) = 0$ , and

$$T(y) = \begin{cases} w(xy), & \text{if } y \in \Gamma^+(x) \\ \infty, & \text{if } y \in V(D) \setminus \{\Gamma^+(x) \cup \{x\}\} \end{cases}$$

Then, choose  $y \in \Gamma^+(x)$  with  $T(y)$  minimum, and set  $P(y) = T(y)$ .

**2** Now, suppose  $z \in V(D)$  is the most recent vertex to receive a permanent label, that is, to have  $P(z)$  defined.  $\forall v \in \Gamma^+(z)$  s.t.  $T(v)$  but not  $P(v)$  is defined, update

$$T(v) = \min(T(v), P(z) + w(zv))$$

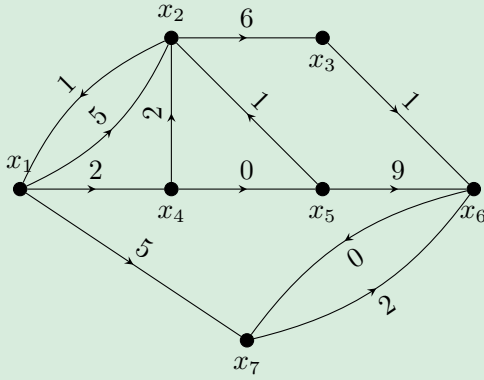
Then  $\forall u \in V(D)$  s.t.  $T(u)$  but not  $P(u)$  is defined, choose  $y$  where  $T(y)$  is minimum, and set  $P(y) = T(y)$ .

**3** Return to Step 2. Repeat procedure until termination.

**Remark 3.1.** • In Step 1 and 2 if there is a tie for the minimum, we may not choose any such suitable  $y$ .

- The assumption  $w : A(D) \rightarrow \mathbb{R}_{\geq 0}$  implies that any shortest path from  $x$  is in fact a path.

**Example 3.1.** Find the shortest paths from  $x_1$  to all other vertices, where  $(D, w)$  is:



We initially set

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
<u>0*</u>	5	$\infty$	2 <sup>*</sup>	$\infty$	$\infty$	5

where    means the out-neighbour of this vertex ( $z$  in Step 2) are being considered, and  $*$  is a permanent label, with <sup>\*</sup> the newest one. The next iteration is

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
<u>0*</u>	4	$\infty$	<u>2*</u>	2 <sup>*</sup>	$\infty$	5

New temporary labels:

$$x_2 : \min(5, 2 + 2) = 4$$

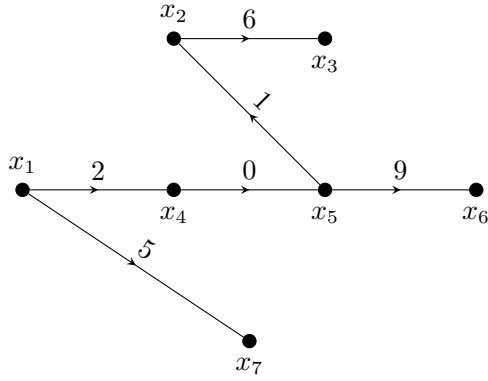
$$x_5 : \min(\infty, 2 + 0) = 2$$

$2 = \min(4, 2, 5)$ , this becomes permanent.

Repeating, we obtain:

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$0^*$	$\boxed{3^*}$	$\infty$	$2^*$	$\underline{2^*}$	5	5
$0^*$	$\underline{3^*}$	9	$2^*$	$2^*$	$\boxed{5^*}$	5
$0^*$	$3^*$	9	$2^*$	$2^*$	$\underline{5^*}$	$\boxed{5^*}$
$0^*$	$3^*$	$\boxed{9^*}$	$2^*$	$2^*$	$5^*$	$\underline{5^*}$

We obtain a shortest paths tree from  $x_1$  as:



To obtain a shortest path from  $x_1$  to  $y$ , look at when  $y$  is given  $\boxed{*}$ , and backtrack.

**Theorem 3.1.** Let  $(D, w)$  be a network digraph, where  $w : A(D) \rightarrow \mathbb{R}_{\geq 0}$ . Then Dijkstra's Algorithm solves Problem 3.1.

*Proof.* Let  $R_t \subset V(D)$  be the set of permanently labelled vertices after  $t$  iterations of Dijkstra's Algorithm. Note that  $R_t = R_{t-1} \cup \{y\}$ , where  $y$  becomes permanently labelled at the  $t^{\text{th}}$  iteration ( $1 < t \leq |V(D)|$ ). We use induction on  $t$  to show that  $P(z) = W(z)$ ,  $\forall z \in R_t \setminus \{x\}$ , where  $W(z)$  is the weight of a shortest path from  $x$  to  $z$ .

**Base cases:**  $t = 1 : R_1 = \{x\}$ , not  $\exists z \in R_1 \setminus \{x\}$ .

$t = 2$ : Let  $R_2 = \{x, z\}$ . Then clearly,  $w(xz) \leq \text{weight of any } x - z \text{ walk}$  since  $w : A(D) \rightarrow \mathbb{R}_{\geq 0}$ .

**Induction Step** Let  $t \geq 3$ , and suppose  $P(z) = W(z)$ ,  $\forall z \in R_{t-1} \setminus \{x\}$ . Let  $R_t = R \cup \{y\}$ . We show that  $P(y) = W(y)$ . Clearly,  $P(y) \geq W(y)$ . Suppose that  $P(y) > W(y)$ . Let  $v_0 v_1 \cdots v_k$  be a shortest path from  $x$  to  $y$ . ( $v_0 = x, v_k = y$ ) with  $\sum_{i=1}^k w(v_{i-1} v_i) = W(y)$ . Since  $y \in R_{t-1}$ ,  $\exists 0 \leq l \leq k$  s.t.  $v_i \in R_{t-1}$ ,  $\forall 0 \leq i \leq l$ , and  $v_{l+1} \notin R_{t-1}$ , then

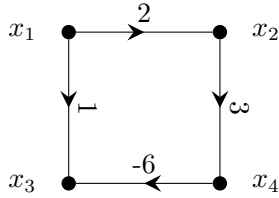
$$P(y) > \sum_{i=1}^k w(v_{i-1} v_i) \geq \sum_{i=1}^k w(v_{i-1} v_i) + w(v_l v_{l+1}) \geq P(v_l) + w(v_l v_{l+1}) \geq T(v_{l+1}) \geq P(y)$$

where  $T(v_{l+1})$  is the temporary label of  $v_{l+1}$  at the  $t^{th}$  iteration. To see (\*), note that at some iteration before  $t$ ,  $v_l$  was given the permanent label  $P(v_l)$ , and  $v_{l+1}$  was given a temporary label  $\leq P(v_l) + w(v_l v_{l+1})$ .  $+$  holds since at  $t^{th}$  iteration,  $y$  is given a permanent label over  $v_{l+1}$  or  $y = v_{l+1}$ .

We have a contradiction. The induction step holds. The proof follows by letting  $R_t = V(D)$ .  $\square$

### Bellman-Ford Algorithm

Now, let  $w : A(D) \rightarrow \mathbb{R}$  in Problem 3.1, so arcs of  $D$  may have negative weights. Then Dijkstra's Algorithm fails. A major problem is if  $D$  has a negative dicycle  $\vec{C} = v_1 v_2 \cdots v_k v_1$  ( $k \geq 2$ ), where  $\sum_{i=1}^k w(v_i v_{i+1}) < 0$  ( $v_{k+1} = v_1$ ). Then no shortest path exists from  $x$  to any  $v_i$ , since we may reach  $v_i$  from  $x$ , then go around  $\vec{C}$  as much as we like. Dijkstra's Algorithm does not detect this problem, since at each iteration, only vertices without a permanent label are explored. Even if  $D$  has no dicycle, there is still a problem like:



If  $x_1$  is the source vertex, then Dijkstra's Algorithm gives  $P(x_3) = 1$ . But the shortest path from  $x_1$  to  $x_3$  is  $x_1 x_2 x_4 x_3$ , with weight  $-1$ . Then proof Th 3.2 breaks down at the base case  $t = 2$ , and also at the induction step.

The **BELLMAN-FORD ALGORITHM** can overcome these problems, and detect the presence of a negative dicycle, but takes more work than Dijkstra's Algorithm. As we run the algorithm, each vertex  $y \in V(D)$  has its label  $L(y)$  updated, which is the currently known weight of an  $x - y$  walk. The algorithm is as follows:

- 1 Fix an ordering of the arcs of  $D$ , say  $a_1, \dots, a_m$ , where  $m = |A(D)|$ .
- 2 Set  $L(x) = 0$  and  $L(y) = \infty$ ,  $\forall y \in V(D) \setminus \{x\}$ .
- 3 In one round, we run through  $a_1, \dots, a_m$ . When we consider  $a_i = uv$  say, update  $L(v)$  by

$$L(v) = \min(L(v), L(u) + w(uv))$$

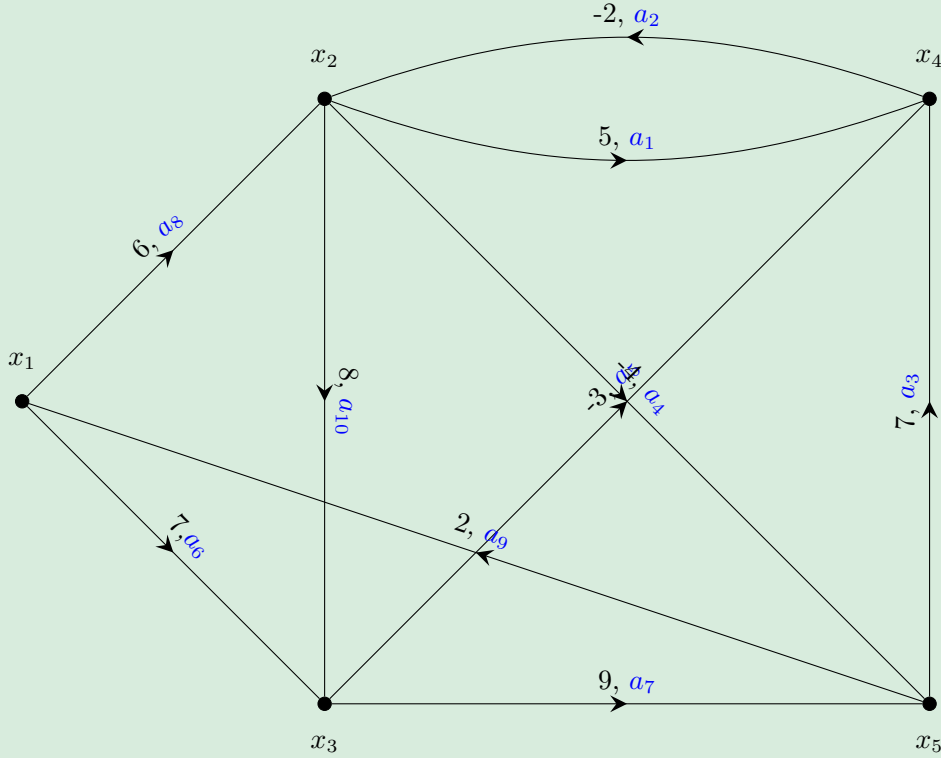
- 4 Repeat Step 3. If after  $\leq |V(D)| - 1$  rounds,  $L(y)$  does not decrease  $\forall y \in V(D)$ , then the resulting  $L(y)$  are the weights of these shortest paths from  $x$  to every  $y \in V(D) \setminus \{x\}$ . Otherwise,  $\exists$  negative dicycle in  $D$ .

Note that:

- After  $r$  rounds step 3, all walks from  $x$  with  $\leq r$  arcs are found.

- If at round  $|V(D)|$ , some  $L(y)$  is updated, then an  $x$ - $y$  walk with  $\geq |V(D)|$  arcs and weight  $L(y)$  has been found. Such a walk contains a dicycle, and then can only exist if  $D$  has a negative dicycle.

**Example 3.2.** Find the shortest paths from  $x_1$  to all others vertices when  $(D, w)$  is the following:



Fix the arc ordering  $a_1, \dots, a_{10}$  as shown. In round 1, we have

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Initially	0	$\infty$	$\infty$	$\infty$	$\infty$
After $a_1, \dots, a_6$	0	$\infty$	7	$\infty$	$\infty$
After $a_7$	0	$\infty$	7	$\infty$	16
After $a_8$	0	6	7	$\infty$	16
After $a_9, a_{10}$	0	6	7	$\infty$	16

Table 3.1: Round 1

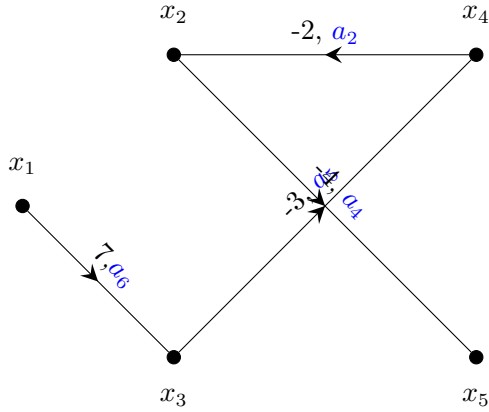
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Initially	0	6	7	$\infty$	16
After $a_1$	0	6	7	11	16
After $a_2, a_3, a_4$	0	6	7	11	2
After $a_5$	0	6	7	4	2
After $a_6, \dots, a_{10}$	0	6	7	4	2

Table 3.2: Round 2

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	
Initially	0	6	7	4	2	
After $a_1, a_2$	0	2	7	4	2	$a_2$
After $a_3, a_4$	0	2	7	4	-2	$a_4$
After $a_5, \dots, a_{10}$	0	2	7	4	-2	

Table 3.3: Round 3

We can then show that round 4 does not update the bottom row of 3.3. The weights of the shortest paths from  $x_1$ , are given by the bottom row of 3.3. We can easily backtrack to obtain the shortest paths tree from  $x_1$ .



For example, for  $x_5$ , see the circle entries. The shortest path has arcs  $a_6, a_5, a_2, a_4$ .

**Theorem 3.2.** Let  $(D, w)$  be a network digraph, where  $w : A((D)) \rightarrow R$ . Suppose we apply the Bellman-Ford Algorithm to solve Problem 3.1.

- (a) Suppose for some  $1 \leq r \leq |V(D)| - 1$ , when going from round  $r$  to round  $r + 1$ , the algorithm does not update  $L(y)$ ,  $\forall y \in V(D)$ . Then  $L(y)$  after round  $r$  is the weight of the shortest path from  $x$  to  $y$ ,  $\forall y \in V(D) \setminus \{x\}$ , and  $L(x) = 0$  at all times.
- (b) If when going from round  $|V(D)| - 1$  to  $|V(D)|$ , the algorithm updates  $L(y)$  for some  $y \in V(D)$ , then  $\exists$  negative dicycle  $\vec{C} \subset D$  ( $|V(\vec{C})| \geq 2$ ). Moreover, for  $z \in V(D) \setminus \{x\}$ ,  $\exists$  shortest path (with finite weight) from  $x$  to  $z \iff z$  is not reachable from any vertex of a negative dicycle of  $D$ .

*Proof.* (a) Note that no  $L(y)$  is updated when we go from round  $s$  to round  $s + 1$ ,  $\forall s \geq r$ . Suppose  $\exists z \in V(D) \setminus \{x\}$ , s.t. at round  $s$ ,  $\exists s \geq r$ , we have  $L(z) > W(z)$ , where

$$W(z) = \begin{cases} \text{Minimum weight of an } x - z \text{ walk,} & \text{if finite,} & (1) \\ -\infty, & \text{otherwise} & (2) \end{cases}$$

If (1) holds, then  $\exists$   $x$ - $z$  walk with weight  $W(z)$ , which is in fact an  $x$ - $z$  path  $\vec{P}$  (see HW5), so

$|A(\vec{P})| \leq |V(D)| - 1$ . But then  $\vec{P}$  would be realised by the algorithm at round  $|V(D)| - 1$ , when we have  $L(z) \leq W(z) < L(z)$ , a contradiction.

If (2) holds, then  $\exists$  sequence of  $x$ - $z$  walks, whose weights strictly decrease and  $\rightarrow -\infty$ . By running the algorithm for as many rounds as we like, all these weights will be realised, and so  $L(z) \rightarrow -\infty$ , a contradiction.

Finally, if  $L(x) < 0$  after some number of rounds, then  $\exists$  negative dicycle containing  $x$ . Running the algorithm as long as we like, we have  $L(x) \rightarrow -\infty$ , a contradiction.

(b) If  $y = x$ , then proof of last part of (a)  $\implies \exists$  negative dicycle. Now let  $y \neq x$ . If  $\exists v \in V(D) \setminus \{x\}$  s.t.  $\neg \exists$  shortest path from  $x$  to  $v$ . ( $\forall M > 0$ ,  $\exists x - v$  walk with weight  $\leq -M$ ), then  $\exists$  negative dicycle (HW5). Otherwise,  $\forall y \in V(D) \setminus \{x\}$ ,  $\exists$  shortest path from  $x$  to  $y$ , and this is an  $x - y$  path (HW5), with  $\leq |V(D)| - 1$  arcs. Hence, the algorithm cannot update after round  $|V(D)| - 1$ , a contradiction.

Final part: ( $\implies$ ) If  $z$  is reachable from some vertex  $u$  of a negative dicycle  $\vec{C}$ , then we may reach  $u$  from  $x$ , go around  $\vec{C}$  as much as we like, then reach  $z$  from  $u$ . Then  $\neg \exists$  shortest path from  $x$  to  $z$ .

( $\impliedby$ ) If  $\neg \exists$  shortest path from  $x$  to  $z$ , then  $\exists$  negative dicycle  $\vec{C}$ , and  $\exists v \in V(\vec{C})$  s.t.  $z$  is reachable from  $v$  (HW5).  $\square$

### 3.3 Maximum Flow Problem

**Definition 3.3.** A **FLOW NETWORK** is a 4-tuple  $(D, c, s, t)$ , where  $(D, c)$  is a network digraph,  $c : A(D) \rightarrow \mathbb{R}_{\geq 0}$  is the **CAPACITY FUNCTION**, and  $s, t \in V(D)$  are the **SOURCE** and **SINK**, s.t.  $t$  is reachable from  $s$ .

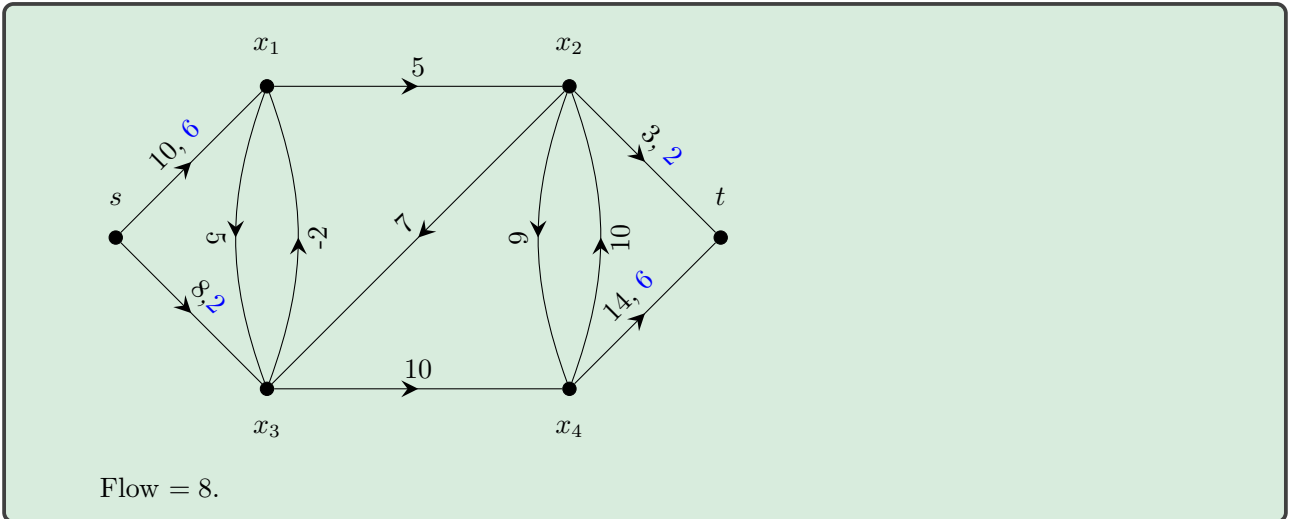
**Definition 3.4.** A **FLOW** of  $(D, c, s, t)$  is  $f : A(D) \rightarrow \mathbb{R}_{\geq 0}$  s.t.  $0 \leq f(a) \leq c(a)$ ,  $\forall a \in A(D)$ , and

$$\sum_{z \in \Gamma^-(x)} f(zx) = \sum_{y \in \Gamma^+(x)} f(xy), \quad \forall x \in V(D) \setminus \{s, t\} \quad (3.2)$$

That is, under  $f$ , the amount flowing into  $x$  and out of  $x$  is same. Note that we always have the flow  $f \equiv 0$ .

**Example 3.3.** A flow network, and a possible flow (reduced capacities in blue):





**Lemma 3.4.** Let  $f$  be a flow of  $(D, s, c, t)$ . Then

$$\sum_{y \in \Gamma^+(s)} f(sy) - \sum_{z \in \Gamma^-(s)} f(zs) = \sum_{z \in \Gamma^-(t)} f(zt) - \sum_{y \in \Gamma^+(t)} f(ty) \quad (3.3)$$

That is, net amount flowing out of  $s$  = net amount flowing into  $t$ . The VALUE  $v(f)$  of  $f$  is the common value of 3.3. In particular, if  $\Gamma^+(s) = \Gamma^-(t) = \emptyset$ , then

$$\sum_{y \in \Gamma^+(s)} f(sy) = \sum_{z \in \Gamma^-(t)} f(zt)$$

*Proof.* Note that

$$\sum_{v \in V(D)} \underbrace{\left( \sum_{y \in \Gamma^+(x)} f(xy) - \sum_{z \in \Gamma^-(x)} f(zx) \right)}_{(*)} = 0$$

since at  $x$ , then terms  $f(xy)$  and  $-f(zx)$  are cancelled by the terms  $-f(xy)$  and  $f(zx)$  from  $y$  and  $z$ . Thus 3.2  $\implies$  The terms  $(*) = 0$ , except when  $x = s, t$  and  $\implies$  3.3  $\square$

By, lemma 3.4, we may ask:

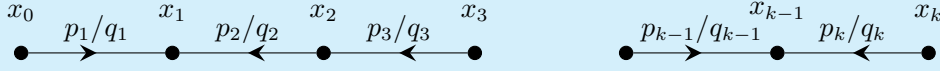
### 3.3.1 Problem 3.5

Given a flow network  $(D, c, s, t)$ , find a maximum flow, that is, a flow  $f$  s.t.  $v(f)$  is maximum.

**Folk-Fulkerson Algorithm**

**Definition 3.5.** Given  $(D, c, s, t)$ , let  $a_1, \dots, a_k \in A(D)$  be s.t. when their directions are ignored, we have a graph path  $x_0x_1 \dots x_k$ , where  $a_i = x_{i-1}x_i$  or  $x_ix_{i-1}$ ,  $\forall 1 \leq i \leq k$ . Assign a symbol  $p_i/q_i$  to  $a_i$  ( $1 \leq i \leq k$ ), where  $p_i + q_i = c(a_i)$ , and  $p_i, q_i \geq 0$ . The  $a_i$  form an  $x_0$ - $x_k$  augmenting path if

$$\begin{cases} p_i > 0, & \text{if } a_i = x_{i-1}x_i : & a_i \text{ is a forward arc} \\ q_i > 0, & \text{if } a_i = x_ix_{i-1} : & a_i \text{ is a backward arc} \end{cases}$$



The Ford-Fulkerson Algorithm for solving Problem 3.5 is as follows.

- Label the arcs of  $D$  as  $a_1, \dots, a_m$  ( $m = |A(D)|$ ). Assign  $c(a_i)/0$  to  $a_i$ ,  $\forall 1 \leq i \leq m$ .
- Suppose  $a_i$  is labelled with  $p_i/q_i$  ( $1 \leq i \leq m$ ). Choose an  $s$ - $t$  augmenting with arcs  $b_1, \dots, b_k$ , labelled with  $p'_i/q'_i$  ( $1 \leq i \leq k$ ). Let

$$r = \min(\min\{p'_i : b_i \text{ is forward}\}, \min\{q'_i : b_i \text{ is backward}\}) > 0$$

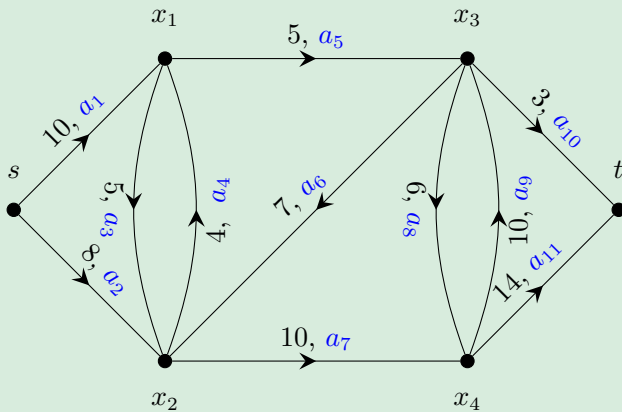
Update  $p'_i/q'_i$  to

$$p'_i/q'_i = \begin{cases} p'_i - r/q'_i + r, & \text{if } b_i \text{ is forward,} \\ p'_i + r/q'_i - r, & \text{if } b_i \text{ is backward.} \end{cases}$$

This directs a flow of  $r > 0$  from  $s$  to  $t$  along the  $s$ - $t$  augmenting path.

- Repeat Step 2 until  $\neg \exists$   $s$ - $t$  augmenting path. The maximum flow is the sum of all the  $r$ 's, achieved by the final  $q_i$ 's

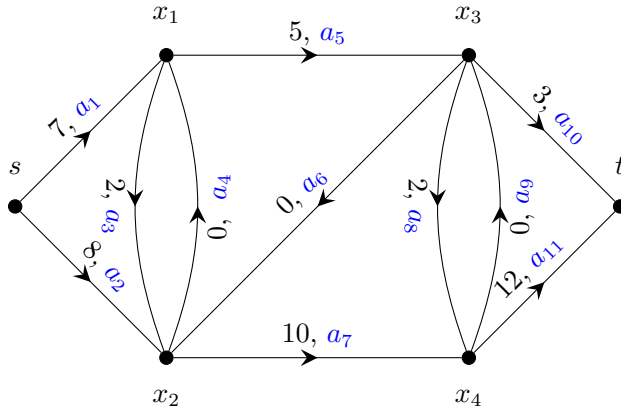
**Example 3'.** Again



We proceed as follows.

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$	$s$ - $t$ augmenting path	flow
<u>10</u> /0	8/0	5/0	4/0	<u>5</u> /0	7/0	10/0	6/0	10/0	<u>3</u> /0	14/0	$a_1 a_5 a_{10}$	3
7/3	<u>8</u> /0	5/0	4/0	2/3	7/0	<u>10</u> /0	6/0	10/0	0/3	<u>14</u> /0	$a_2 a_7 a_{11}$	8
<u>7</u> /3	0/8	5/0	4/0	2/3	7/0	<u>2</u> /8	6/0	2/8	0/3	<u>6</u> /8	$a_1 a_5 a_6 a_7 a_{11}$	2
<u>5</u> /5	0/8	<u>5</u> /0	4/0	0/5	<u>5</u> /2	0/10	<u>6</u> /0	2/8	0/3	<u>4</u> /10	$a_1 a_3 \overleftarrow{a_6} a_8 a_{11}$	2
3/7	0/8	3/2	4/0	0/5	7/0	0/10	4/2	10/0	0/3	2/12		

The bottom row has no  $s$ - $t$  augmenting path. The maximum flow is  $3 + 8 + 2 + 2 = 15$ , achieved as follows:

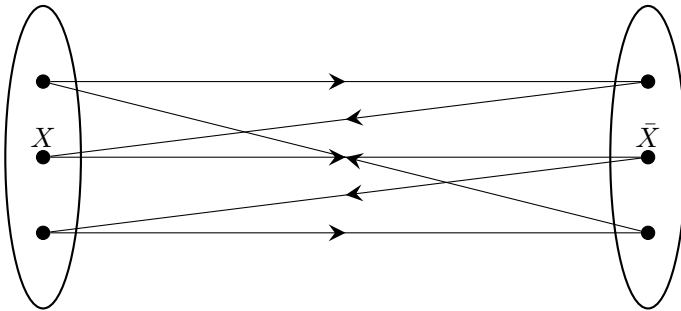


**Definition 3.6.** Let  $(D, c, s, t)$  be a flow network. For disjoint  $X, Y \subset V(D)$ , let

$$A(X, Y) = \{xy \in A(D) : x \in X, y \in Y\},$$

$$c(X, Y) = \sum_{xy \in A(X, Y)} c(xy)$$

A **CUT (SEPARATING  $s$  AND  $t$ )** is a set  $A(X, \bar{X})$ , where  $s \in X$ ,  $t \in \bar{X}$  and  $\bar{X} = V(D) \setminus X$ . Thus by deleting a cut  $A(X, \bar{X})$  from  $D$ , the remaining subdigraph  $D' = (V(D), A(D) \setminus A(X, \bar{X}))$  is s.t.  $t$  is not reachable from  $s$ . A **MINIMUM CUT** of  $(D, c, s, t)$  is a cut  $A(X, \bar{X})$  s.t.  $c(X, \bar{X})$  is minimum.



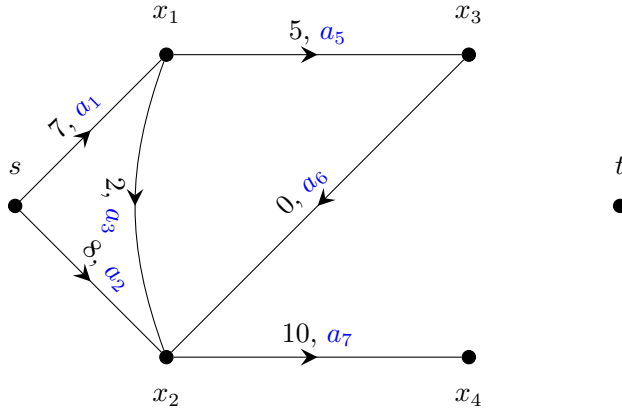
The Ford-Fulkerson Algorithm also finds a minimum cut of  $(D, c, s, t)$ . When the algorithm termi-

nates, let

$$X = \{x \in V(D) : \exists s-x \text{ augmenting path w.r.t. final flow}\} \cup \{s\}$$

Note that  $t \in \bar{X}$ , otherwise  $\exists s-t$  augmenting path when the algorithm terminated, a contradiction. So  $A(X, \bar{X})$  is a cut, and in fact, a minimum cut.

**Example 3'.** After the algorithm terminates in Example 3', we see that  $\exists s-t$  augmenting path  $\iff x \in \{x_1, x_2\}$ .



We have  $X = \{s, x_1, x_2\}$ ,  $\bar{X} = \{t, x_3, x_4\}$ , and  $A(X, \bar{X}) = \{a_5, a_7\}$  with  $c(X, \bar{X}) = 5 + 10 = 15$ .

We have the following result

**Theorem 3.6** (Max-flow min-cut Theorem). For a flow network  $(D, c, s, t)$ , the maximum flow is equal to  $c(X, \bar{X})$ , where  $A(X, \bar{X})$  is a minimum cut.

We first prove:

**Theorem 3.7** (Flow value lemma). Let  $f$  be a flow and  $A(X, \bar{X})$  be a cut of  $(D, c, s, t)$ . Then

$$v(f) = \sum_{xy \in A(X, \bar{X})} f(xy) - \sum_{yx \in A(\bar{X}, X)} f(yx)$$

*Proof.* We have

$$\begin{aligned} v(f) &= \sum_{y \in \Gamma^+(s)} f(sy) - \sum_{z \in \Gamma^-(s)} f(zs) \\ &= \sum_{x \in X} \underbrace{\left( \sum_{y \in \Gamma^+(s)} f(xy) - \sum_{z \in \Gamma^-(s)} f(zx) \right)}_{(*)}, \text{ the terms } (*) = 0, \text{ except for } x = s \\ &= \sum_{xy \in A(X, \bar{X})} f(xy) - \sum_{yx \in A(\bar{X}, X)} f(yx), \text{ arcs within } X \text{ cancel.} \end{aligned}$$

□

*Proof.* of Th 3.6. For any flow  $g$  and any cut  $A(Y, \bar{Y})$ , we have  $v(g) \leq c(Y, \bar{Y})$  (HW5). So  $v = \sup\{v(g) : g \text{ is a flow of } (D, c, s, t)\} < \infty$ . We may take a sequence of flow  $g_j : A(D) \rightarrow \mathbb{R}_{\geq 0}$  s.t.  $v(g_j) \rightarrow v$  and  $g_j(xy) \rightarrow f(xy)$ ,  $\forall xy \in A(D)$  ( $f$  is the point-wise limit of the  $g_j$ ). Then  $f$  is a minimum flow. Now, we construct a cut  $A(X, \bar{X})$  s.t.  $v(f) = c(X, \bar{X})$ . Define

$$X = \{x \in V(D) : \exists s-x \text{ augmenting path w.r.t. } f\} \cup \{s\}$$

We claim that  $A(X, \bar{X})$  is a cut. Otherwise, suppose  $t \in X$ . Then  $\exists s = x_0, x_1, \dots, x_k = t \in V(D)$  s.t.

$$r_i = c(x_{i-1}x_i) - f(x_{i-1}x_i) \text{ or } r_i = f(x_i x_{i-1})$$

satisfies  $r_i > 0$ ,  $\forall 1 \leq i \leq k$ . Let  $r = \min_i r_i > 0$ . We define the flow  $f^*$  by

$$\begin{aligned} f^*(x_{i-1}x_i) &= f(x_{i-1}x_i) + r \leq c(x_{i-1}x_i), & \text{if } r_i &= c(x_{i-1}x_i) - f(x_{i-1}x_i) \\ f^*(x_{i-1}x_i) &= f(x_i x_{i-1}) - r \geq 0, & \text{if } r_i &= f(x_i x_{i-1}), \\ f^*(a) &= f(a), & \forall \text{ other arcs } a. \end{aligned}$$

Then  $f^*$  is also a flow, and  $v(f^*) = v(f) + r$  since either  $sx_1$  increased by  $r$ , or  $x_1s$  decreased by  $r$ . This contradicts that  $f$  is a maximum flow.

Now,

$$v(f) = \text{Th 3.7} \sum_{xy \in A(X, \bar{X})} \underbrace{f(xy)}_{=c(xy)} - \sum_{yx \in A(X, \bar{X})} \underbrace{f(yx)}_{=0} = c(X, \bar{X}),$$

as required. □

**Theorem 3.8.** The Ford-Fulkerson Algorithm, if it terminates, solves Problem 3.5 for  $(D, c, s, t)$  and also finds a minimum cut. Precisely, let  $f$  be the final flow, and

$$X = \{x \in V(D) : \exists s-x \text{ augmenting path w.r.t } f\} \cup \{s\}$$

Then  $f$  is a maximum flow, and  $A(X, \bar{X})$  is a minimum cut.

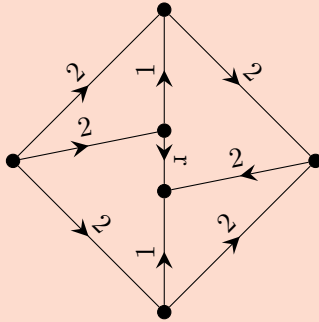
*Proof.* From the discussion before Example 3",  $A(X, \bar{X})$  is a cut. Note that if  $xy \in A(X, \bar{X})$  then  $f(xy) = c(xy)$ , and if  $yx \in A(X, \bar{X})$  then  $f(yx) = 0$ . Otherwise, in either case,  $\exists s-y$  augmenting path:

Take an  $s - x$  augmenting path, which must be within  $X$ , then go along  $xy$  or  $yx$  to  $y$ . Thus

$$v(f) = \sum_{xy \in A(X, \bar{X})} \underbrace{f(xy)}_{=c(xy)} - \sum_{yx \in A(\bar{X}, X)} \underbrace{f(yx)}_{=0} = c(X, \bar{X})$$

If  $A(Y, \bar{Y})$  is a minimum cut, then  $c(X, \bar{X}) = v(f) \leq c(Y, \bar{Y})$  (HW5). So  $A(X, \bar{X})$  is a minimum cut, and Th 3.6  $\Rightarrow$   $f$  is a maximum flow.  $\square$

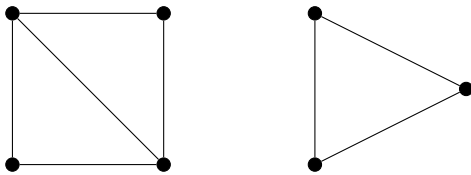
**Remark 3.2.** The Ford-Fulkerson Algorithm terminates if  $c : A(D) \rightarrow \mathbb{Z}_{\geq 0}$ , since each iteration increase the currently known flow by  $\geq 1$ . The algorithm also terminates if  $c : A(D) \rightarrow \mathbb{Q}_{\geq 0}$ , since we may consider  $c' : A(D) \rightarrow \mathbb{Z}_{\geq 0}$  by multiplying the denominators of  $c(xy)$ ,  $\forall xy \in A(D)$  by a common multiple. But the algorithm may possible not terminate if  $c : A(D) \rightarrow \mathbb{R}_{\geq 0}$ . For example:



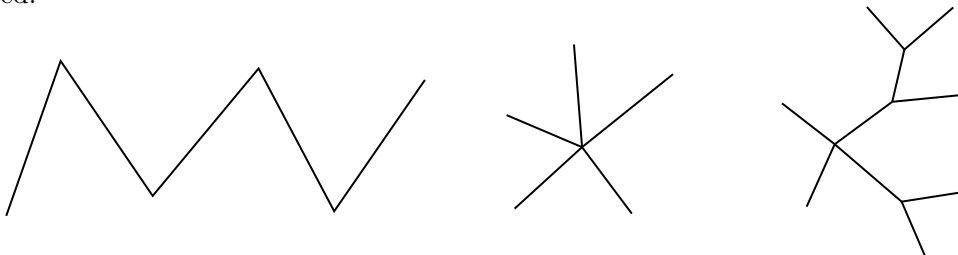
$r = \frac{\sqrt{5}-1}{2}$ , which satisfies  $r^2 = 1 - r$ .

### 3.4 Minimum Spanning Tree Problem

**Definition 3.7.** A (connected) component of a graph  $G$  is a “maximal connected subgraph”  $H \subset G$  in the following sense:  $H$  is connected, and  $\neg \exists xy \in E(G) \setminus E(H)$  s.t.  $H' = (V(H) \cup \{x, y\}, E(H) \cup \{xy\})$  is connected.



A graph  $T$  is a tree if  $T$  is connected; and does not contain a cycle as a subgraph. A graph  $F$  is a forest if all components of  $F$  are trees. A leaf of a graph is a vertex with degree 1. Trees, with leaves circled.



### 3.4.1 Problem 3.9 (Minimum Spanning Tree Problem)

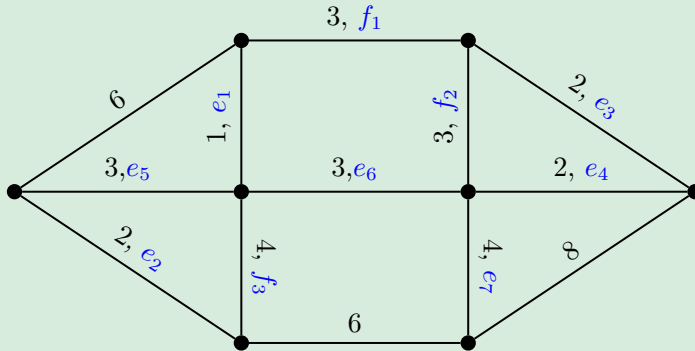
Let  $(G, W)$  be a network graph, where  $G$  is connected and  $w : E(G) \rightarrow \mathbb{R}$ . Find a minimum spanning tree (MST), that is, a spanning tree (ST)  $T$  of  $G$  s.t.  $\sum_{e \in E(T)} w(e)$  is minimum.

Kruskal's Algorithm can solve Problem 3.9, and is as follows.

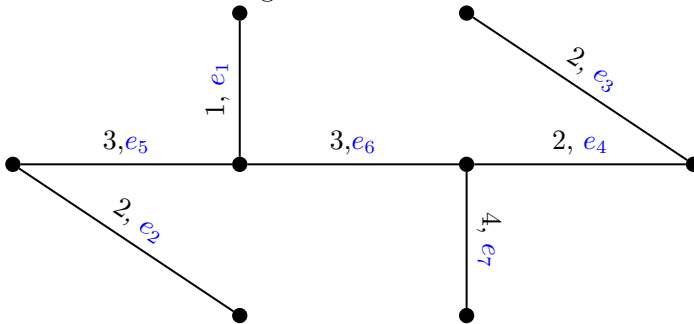
#### Kruskal's Algorithm

- Choose an edge  $e_1 \in E(G)$  with  $w(e_1)$  minimum.
- Suppose edges  $e_1, \dots, e_k \in E(G)$  have been chosen, for some  $k \geq 1$ . Choose  $e_{k+1} \in E(G) \setminus \{e_1, \dots, e_k\}$  s.t.  $e_{k+1}$  does not form a cycle with some of  $e_1, \dots, e_k$  and  $w(e_{k+1})$  is minimum.
- Continue until  $|V(G)| - 1$  edges are found. The result is a MST.

**Example 4'.** Let  $(G, W)$  be



We “greedily” choose  $e_1, \dots, e_6$ , since they have the smallest weights, and do not form a cycle. Then we cannot choose  $f_1$  or  $f_2$  since adding either edge forms a cycle. We can then choose  $e_7$  (but not  $f_3$ ) to obtain the MST. The weight is  $1 + 2 + 2 + 2 + 3 + 3 + 4 = 17$ .



**Theorem 3.9.** Kruskal's Algorithm solves Problem 3.9.

*Proof.* Let  $|V(G)| = n$ . We first show that when the algorithm terminates, we have a ST  $T$  of  $G$ .

Initially, we have a forest with  $n$  single vertices as components. Suppose that after  $t$  iterations ( $0 \leq t \leq n - 2$ ), the current graph  $F$  is a forest with  $n - t$  components. If we add a new edge  $xy$  to  $F$  to form  $F^+$ , then we cannot have  $x, y$  belonging to the same component of  $F$ , otherwise we create a

cycle. Also, since  $G$  is connected, we may add  $xy$  where  $x, y$  belong to two different components  $F_1, F_2$  of  $F$ . The new component  $F' = (V(F_1) \cup V(F_2), E(F_1) \cup E(F_2) \cup \{xy\})$  of  $F^+$  is also a tree. Clearly  $F'$  is connected, and if  $\exists$  cycle  $C \subset F'$ , then  $\exists e, e' \in E(C)$ , each one connecting  $F_1$  and  $F_2$ , a contradiction since  $xy$  is the only such edge in  $F'$ . Thus,  $F^+$  is a forest with  $n - t - 1$  components. Iterating  $n - 1$  times gives a ST T of  $G$ .

Now, we prove that  $T$  is a MST. Let  $e_1, \dots, e_{n-1}$  chosen in that order, so  $w(e_1) \leq \dots \leq w(e_{n-1})$ . If  $T$  is not a MST, choose a MST  $S \subset G$  s.t.  $|E(S) \cap E(T)|$  is maximum. Let  $1 \leq k \leq n - 1$  be s.t.  $e_1, \dots, e_{k-1} \in E(S) \cap E(T)$ , and  $e_k \in E(T) \setminus E(S)$ . Then  $S' = (V(S), E(S) \cup \{e_k\})$  has a cycle  $C'$ . Now,  $\exists f \in E(C') \setminus E(T)$ . Then  $S'' = (V(S'), E(S') \setminus \{f\})$  is also a ST, since  $S''$  is connected and has  $n - 1$  edges (HW6). To see that  $S''$  is connected, let  $u, v \in V(S'')$ . Then  $\exists u - v$  path  $P \subset S'$ . Either  $P \subset S''$ , or  $f \in E(P)$ . If the latter, let  $f = zz'$ . Then  $\exists u - z$  and  $z' - v$  paths  $\subset P$ , and a  $z - z'$  path in  $C'$ , not using  $f$ . These paths  $\implies \exists u - v$  path in  $S''$ .

Now  $\sum_{e \in E(S'')} w(e) \geq \sum_{e \in E(S)} w(e) \implies w(e_k) \geq w(f)$ . Since  $e_1, \dots, e_{k-1}, e_k \in E(T)$  and  $e_1, \dots, e_{k-1}, f \in E(S)$ , neither of these sets of edges creates a cycle, and since the algorithm chose  $e_k$  instead of  $f$ , we have  $w(e_k) \geq w(f)$ . So  $w(e_k) = w(f)$ , and  $S''$  is also a MST. But then  $|E(S'') \cap E(T)| = |E(S) \cap E(T)| + 1$ , which contradicts the choice of  $S$ .  $\square$

**Remark 3.3.** • Kruskal's Algorithm also work if some edges of  $G$  has negative weights.

- If there is a tie for the choice of edges with minimum weight at an iteration, we may choose any suitable edge.
- If  $G$  has  $q$  components, then Kruskal's Algorithm finds a minimum spanning forest in  $n - q$  iterations.

**Proposition 3.11.** Let  $T$  be a tree with  $n \geq 2$  vertices.

- (a)  $T$  has a leaf.
- (b) Deleting a leaf  $x$  and the edge  $e$  incident to  $x$  from  $T$  gives another tree  $T'$ .

*Proof.* True for  $n = 2$ . Assume  $n \geq 3$ .

- (a) Let  $P = x_0x_1 \dots x_k \subset T$  be a longest path, for some  $k \geq 2$ . Then  $\Gamma(x_k) \subset \{x_0, \dots, x_{k-1}\}$ , otherwise we have a path longer than  $P$ . Also,  $x_kx_i \notin E(T)$ ,  $\forall 0 \leq i \leq k - 2$ , otherwise  $T$  has a cycle. So  $\Gamma(x_k) = x_{k-1}$ , and  $x_k$  is a leaf.
- (b) Clearly  $\nexists$  cycle in  $T \implies \nexists$  cycle in  $T'$ . Let  $u, v \in V(T') = V(T) \setminus \{x\}$ . Then  $\exists u - v$  path  $Q \subset T$ , and  $Q$  cannot use  $e$  and  $x$  since  $d(x) = 1$  in  $T$ . So  $G \subset T'$  and  $T'$  is connected.

$\square$



Prop 3.11 is very useful for induction proofs involving trees.



# Chapter 4

## Queueing Theory

### 4.1 Queueing system

Queueing happens in everyday life. At the supermarket or airport, bank and so on. A typical queueing system (say at the airport check-in):

Suppose the  $n^{th}$  customer  $C_n$  arrives at time  $\tau_n \geq 0$ , where  $\tau_n = 0$  and  $0 \leq \tau_1 \leq \tau_2 \leq \dots \leq$ . The interarrival time between customers  $c_n$  and  $c_{n+1}$  is  $T_n = \tau_{n+1} - \tau_n$  ( $T_0 = \tau_1$ ). The service time for customer  $C_n$  is  $X_n \geq 0$ . Assume the  $T_n$  are independent random variables, and the same for the  $X_n$ . The distribution function (DF) of  $T_n$  ( $n \geq 0$ ) is  $A_n : \mathbb{R} \rightarrow [0, 1]$ , where

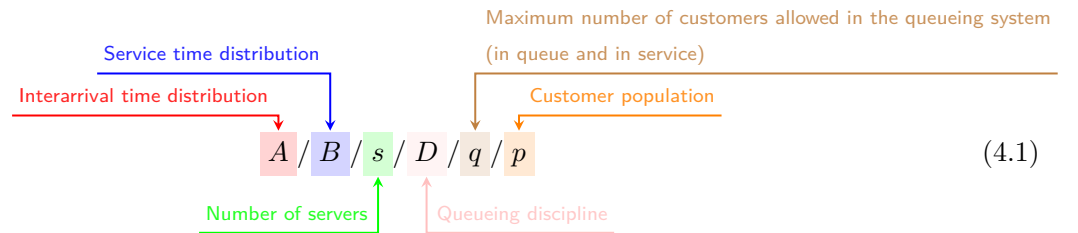
$$A_n(t) = \mathbb{P}(T_n \leq t)$$

and the probability density function (PDF) is

$$a_n(t) = A'_n(t)$$

Similarly,  $X_n$  ( $n \geq 1$ ) has DF  $B_n(t) = \mathbb{P}(X_n \leq t)$  and PDF  $b_n(t) = B'_n(t)$ .

**Kendall-Lee notation** A queueing system description is written as:



**Notable cases:** For  $A$  and  $B$ :

- $M$ : Markovian, with exponential distribution

$$A_n(t) = 1 - e^{-\lambda_n t}, \quad a_n(t) = \lambda_n e^{-\lambda_n t}$$

for some parameter  $\lambda_n > 0$  ( $n \geq 0$ ). Similarly for  $B_n(t)$ ,  $b_n(t)$  with parameter  $\mu_n > 0$  ( $n \geq 1$ ).

- $D$ : Deterministic, with “constant distribution”.
- $E_k$ : Erlang-k
- $G$ : General, not specified, usually the mean and variance are known.

For  $D$ :

- FCFS: First come first served. Services is performed according to the order of the queue. New arrival join the back of the queue.
- LCFS: Last come first served.
- SIRO: Service in random order.

**Note also** Nucleic acid test queueing discipline.

Arrival order: 1, 2,  $\dots$ , 20, 21, 22,  $\dots$ , 40, 41,  $\dots$

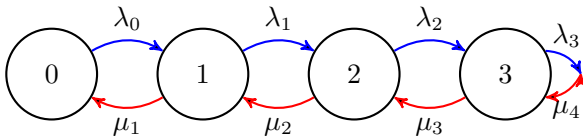
Service order: 20, 19,  $\dots$ , 40, 39,  $\dots$ , 21, 60,  $\dots$

For example,  $M/G/2/SIRO/50/1000$ . If  $D = FCFS$ ,  $q = p = \infty$ , then we write 4.1 as  $A/B/s$ . The simplest case is  $M/M/s$ , especially  $M/M/1$ .

**Definition 4.1.** • State/Queue length = Number of customers in the queueing system/queue (12 and 8 in the diagram).

- $N(t)$  = State at time  $t \geq 0$ .
- $\mathbb{P}_k(t)$  = Probability of state  $k$  at time  $t \geq 0$  (assume  $\mathbb{P}_k(0) = 0$ ).
- $\lambda_k$  = Mean arrival rate = Expected number of new customer arrivals per unit time, when state =  $k \geq 0$ .
- $\mu_k$  = Mean service rate for overall system = Expected number of customers served per unit time between all busy servers (those serving customers), when state =  $k \geq 1$ .

When  $A$  and  $B$  are both Markovian, we have the birth-death process rate diagram:



If  $\lambda_k$  is constant  $\forall k \geq 0$ , write  $\lambda_k = \lambda$ . If the mean service rate per busy server is constant  $\forall k \geq 1$ , denote this constant by  $\mu$ . Then

$$\mu_k = \begin{cases} k\mu & 1 \leq k \leq s \\ s\mu & k > s. \end{cases}$$

Under these circumstance,  $\frac{1}{\lambda}$  and  $\frac{1}{\mu}$  are the expected interarrival time and the expected service time. Also,

$$\rho = \frac{\lambda}{s\mu}$$

is the utilisation factor of the service facility, which is the expected fraction of time the service capacity ( $s\mu$ ) is being utilised by arriving customers ( $\lambda$ ).

**Example 1'.** Suppose a time unit is one hour.

- If on average, a customer arrives every 10 minutes, then  $\lambda = 6$  customers/hour. Expected interarrival time  $\frac{1}{\lambda} = \frac{1}{6}$ .
- If on average, a service time takes 30 minutes, then  $\mu = 2$  customers/hour. Expected service time  $= \frac{1}{\mu} = \frac{1}{2}$ .

Utilisation factor  $\rho = \frac{\lambda}{s\mu} = \frac{3}{s}$ .

- If  $s = 1, 2$ , then  $\rho > 1$ . The state  $\rightarrow \infty$  as time  $\rightarrow \infty$ , and the system is “unsteady”.
- If  $s = 3$ , then  $\rho = 1$ . Queue is unstable and system may not become “steady”.
- If  $s \geq 4$ , then  $\rho < 1$ . The system will become “steady”.

**Definition 4.2.**

$$\begin{array}{c}
 \text{The average state over } [0, t] \\
 \downarrow \\
 \text{The long term average state} \quad \bar{L}(t) = \frac{1}{t} \int_0^t N(u) du \\
 \downarrow \\
 \bar{L} = \lim_{t \rightarrow \infty} \bar{L}(t) \\
 \\
 \begin{array}{c}
 \text{The average number of arrivals in } [0, t] \quad \bar{\alpha}(t) = \frac{\alpha(t)}{t} \\
 \uparrow \\
 \text{The long term average arrival rate} \quad \bar{\lambda} = \lim_{t \rightarrow \infty} \bar{\alpha}(t)
 \end{array}
 \end{array}
 \quad \begin{array}{c}
 \text{Number of arrivals in } [0, t] \\
 \downarrow \\
 \alpha(t)
 \end{array}
 \quad (4.2)$$

- Sojourn time = Time a customer spends in queueing system.

- $W_k$  = Sojourn time of customer  $C_k$ .
- $\bar{W}(t) = \frac{1}{\alpha(t)} \sum_{k=1}^{\alpha(t)} W_k$ , the average sojourn time in  $[0, t]$ .

$$\bar{W} = \lim_{t \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n W_k \quad (4.3)$$

the long term average sojourn time

**Theorem 4.1** (Little' Low). For any queueing system,

$$\bar{L} = \bar{\lambda} \bar{W}, \quad (4.4)$$

provided the limits 4.2, 4.3 exist.

*Proof.* Let  $\beta(t)$  be the number of departures in  $[0, t]$ . We compare the graphs of  $\alpha(t)$ ,  $\beta(t)$ , and the sojourn times.

Let  $E(t)$  be the excess sojourn time of all active customers beyond time  $t$ . Then

$$\begin{aligned} \bar{L}(t) &= \frac{1}{t} \int_0^t N(u) du = \frac{1}{t} \int_0^t (\alpha(u) - \beta(u)) du \\ &= \frac{\alpha}{t} \frac{1}{\alpha} \left( \sum_{k=1}^{\alpha(t)} W_k - E(t) \right) = \bar{\alpha}(t) \bar{W}(t) - \frac{E(t)}{t} \end{aligned}$$

Letting  $t \rightarrow \infty$  and assuming  $\frac{E(t)}{t} \rightarrow 0$ , 4.4 follows.  $\square$

**Remark 4.1.** If  $\bar{L}_q$  = Long term average queue length,  $\bar{W}_q$  = Long term average queueing time, then we can similarly prove that

$$\bar{L}_q = \bar{\lambda} \bar{W}_q$$

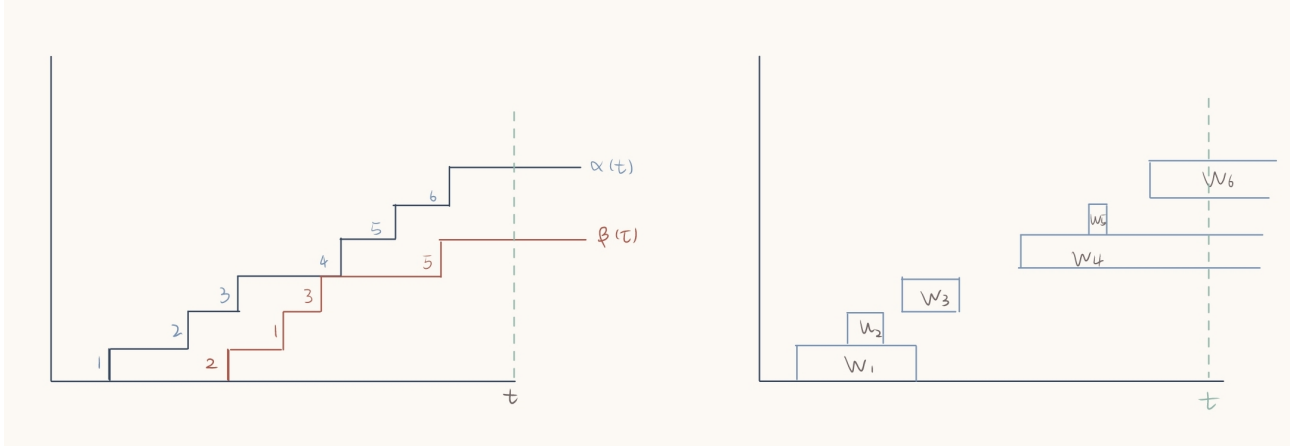
Also, if the expected service time  $\frac{1}{\mu}$  is constant, then

$$\bar{W} = \bar{W}_q + \frac{1}{\mu}$$

**Example 2'.** Bob owns a small coffee shop. He estimates that on average, in one hour, 40 customers arrive at his shop, and each customer spends 6 minutes in the queueing system. Then the average

number of customers in the queueing system is

$$\bar{L} = \underbrace{40}_{\bar{\lambda}} \times \underbrace{0.1}_{\bar{w}} = 4$$



## 4.2 The exponential distribution

Recall that in a Markovian queueing system.  $A/B/s$  both  $A = M$ (interarrival time) and  $B = M$ (service time) follow the exponential distributions, with DF and PDF of the form

$$F(t) = \mathbb{P}(T \leq t) = \begin{cases} 1 - e^{-\gamma t} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases}$$

$$f(t) = \begin{cases} \gamma e^{-\gamma t} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases}$$

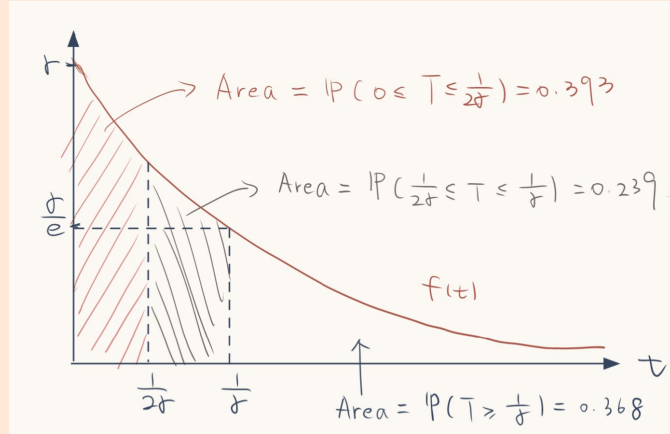
for some  $\gamma > 0$ , where  $T$  is the  $A$  or  $B$  RV. We have

$$\mathbb{E}T = \int_0^\infty \gamma u e^{-\gamma u} du = \frac{1}{\gamma}$$

$$\text{var}T = \mathbb{E}(T^2) - (\mathbb{E}T)^2 = \int_0^\infty \gamma u^2 e^{-\gamma u} du - \frac{1}{\gamma^2} = \frac{1}{\gamma^2}$$

**Proposition 4.2.**  $f(t)$  is strictly decreasing for  $t \geq 0$ . Thus,

$$\mathbb{P}(0 \leq T \leq h) > \mathbb{P}(t \leq T \leq t+h), \quad \forall t, h > 0$$



**Recall** For events  $E_1, E_2$  in a probability space, the conditional probability of  $E_1$ , given  $E_2$ , is

$$\mathbb{P}(E_1 | E_2) = \frac{\mathbb{P}(E_1 \cap E_2)}{\mathbb{P}(E_2)}$$

**Proposition 4.3** (Lack of memory).

$$\mathbb{P}(T > t + h | T > h) = \mathbb{P}(T > t) \quad \forall t, h > 0$$

That is, the probability distribution of the remaining time until the event (arrival or service completion) occurs is always the same, regardless of how much time  $h$  has already passed. There is no memory of what has already occurred.

*Proof.*

$$\begin{aligned} \mathbb{P}(T > t + h | T > h) &= \frac{\mathbb{P}(T > t + h \text{ and } T > h)}{\mathbb{P}(T > h)} = \frac{\mathbb{P}(T > t + h)}{\mathbb{P}(T > h)} \\ &= \frac{e^{-\gamma(t+h)}}{e^{-\gamma h}} = e^{-\gamma t} = \mathbb{P}(T > t) \end{aligned}$$

□

**Proposition 4.4.** Let  $T_1, \dots, T_n$  be independent exponential RVs with parameters  $\gamma_1, \dots, \gamma_n$ . Then the RV

$$U = \min(T_1, \dots, T_n)$$

has an exponential distribution with parameter  $\gamma = \gamma_1 + \dots + \gamma_n$ . Thus, if  $T_i$  represents the time until the  $i^{\text{th}}$  event occurs, then  $U$  represents the time until the first of the  $n$  events occurring.



*Proof.*

$$\begin{aligned}
 \mathbb{P}(U > t) &= \mathbb{P}(T_1 > t, \dots, T_n > t \text{ all hold}) \\
 &= \mathbb{P}(T_1 > t) \cdots \mathbb{P}(T_n > t) \\
 &= e^{-\gamma_1 t} \cdots e^{-\gamma_n t} = e^{-\gamma t}
 \end{aligned}$$

By independence

□

In particular, if  $X_1, \dots, X_n$ s are the independence service times, each exponential with parameter  $\mu$ , then  $U$  is the time until the next service completion, which is the exponential with parameter  $s\mu$ .

### Relationship with Poisson distribution and process

Suppose the time between consecutive occurrences of some kind of event (For example, interarrivals) has an exponential distribution with parameter  $\gamma$ . Let  $Y(t)$  be the number of event occurrences at time  $t \geq 0$ . Then  $Y(t)$  is a RV with the Poisson Distribution with parameters  $\gamma t$ , given by

$$\mathbb{P}(Y(t) = k) = \frac{(\gamma t)^k e^{-\gamma t}}{k!}, \quad \text{for } k = 0, 1, 2, \dots$$

When the events are counted on a continuing basis, the counting process  $\{Y(t) : t \geq 0\}$  is a Poisson process with mean rate  $\gamma$ .

#### Proposition 4.5.

$$\mathbb{P}(T \leq t + h \mid T > t) \approx \gamma h, \quad \forall t > 0 \text{ and small } h$$

*Proof.*

$$\begin{aligned}
 \mathbb{P}(T \leq t + h \mid T > t) &= \frac{\mathbb{P}(T \leq t + h \text{ and } T > t)}{\mathbb{P}(T > t)} \\
 &= \frac{(1 - e^{-\gamma(t+h)}) - (1 - e^{-\gamma t})}{e^{-\gamma t}} \\
 &= 1 - e^{-\gamma h} \approx \gamma h \text{ for small } h
 \end{aligned}$$

□

Note that Prop 4.3 says that the probability of an event occurring within an interval of fixed length  $h$  is constant, regardless of how much time has passed. Prop 4.5 says that if  $h$  is small, then this constant probability is approximately  $\gamma h$ .

### 4.3 Birth-Death Process and M/M/s Systems

Recall that  $\mathbb{P}_k(t)$  = Probability that state =  $k \geq 0$  at time  $t \geq 0$  in a queueing system.

**Definition 4.3.** A queueing system is in transient state if  $\mathbb{P}_k(t)$  depends on  $t$ . The system becomes steady state if  $\mathbb{P}_k(t)$  becomes independent of  $t$  if  $t \geq t_1$  for some (large) time  $t_1$ . If the system is in steady state, let

$$p_k = \text{Probability that state} = k \geq 0$$

The birth-death process describe the state of a queueing system over time. If the system is Markovian, we assume the following:

- (1) Given  $N(t) = k$ , the probability distributions of the remaining times until the next “birth” (arrival) and next “death” (departure) are exponential with parameters  $\lambda_k, \mu_k$ .
- (2) All RVs in (1) are mutually independent. The next change in state is

either:  $k \rightarrow k + 1$  (a birth)

or:  $k \rightarrow k - 1$  (a death)

**Remark 4.2.** By Prop 4.3, for the Markovian birth-death process, the future depends only on the current state, and is independent of any past events. The process is thus a continuous time Markov chain.

The relation of the exponential distribution to the Poisson process implies that the  $\lambda_k$  and  $\mu_k$  are mean rates.

Assume that a queueing system can reach steady state. Let  $u_k(t)$  and  $v_k(t)$  be the number of times the birth-death process enters and leave state  $k$  in  $[0, t]$ . Then  $|u_k(t) - v_k(t)| \leq 1$ , so

$$\lim_{t \rightarrow \infty} \left| \frac{u_k(t)}{t} - \frac{v_k(t)}{t} \right| = 0$$

Since  $\frac{u_k(t)}{t}$  and  $\frac{v_k(t)}{t}$  are the mean rates at which the process enters and leaves state  $k$ , we have:

**Proposition 4.6** (Rate in = Rate out Principle). For any state  $k \geq 0$  in a Markovian birth-death Process,

$$\text{Mean entering rate} = \text{Mean leaving rate} \tag{4.5}$$

An equation of the form 4.5 is a balance equation.

For state 0, the process may only enter from state 1. In steady state, since  $p_1$  is the proportion of

time that the process can enter state 0, the mean entering rate to state 0 is  $\mu_1 p_1$ . Similarly, the mean leaving rate from state 0 is  $\lambda_0 p_0$ . Thus 4.5  $\Rightarrow$

$$\mu_1 p_1 = \lambda_0 p_0 \quad (4.6)$$

For any other state  $k \geq 1$ , we similarly have

$$\lambda_{k-1} p_{k-1} + \mu_{k+1} p_{k+1} = (\lambda_k + \mu_k) p_k \quad (4.7)$$

Now, 4.6  $\Rightarrow \mu_1 p_1 - \lambda_0 p_0 = 0$ . And 4.7  $= \mu_{k+1} p_{k+1} - \lambda_k p_k = \mu_k p_k - \lambda_{k-1} p_{k-1}$ ,  $\forall k \geq 1$ . Thus  $\mu_k p_k - \lambda_{k-1} p_{k-1} = 0$ ,  $\forall k \geq 0$ .

We have

$$\begin{aligned} p_k &= \frac{\lambda_{k-1}}{\mu_k} p_{k-1} + \frac{1}{\mu_k} (\mu_{k-1} p_{k-1} - \lambda_{k-2} p_{k-2}) = \frac{\lambda_{k-1}}{\mu_k} p_{k-1} \\ &= \dots = \frac{\lambda_{k-1} \lambda_{k-2} \dots \lambda_0}{\mu_k \mu_{k-1} \dots \mu_0} p_0 \end{aligned}$$

Let

$$r_k = \frac{\lambda_{k-1} \lambda_{k-2} \dots \lambda_0}{\mu_k \mu_{k-1} \dots \mu_0}, \quad \forall k \geq 1, \quad \text{and } r_0 = 1 \quad (4.8)$$

Then  $p_k = r_k p_0$ ,  $\forall k \geq 0$ . Since  $\sum_{k=0}^{\infty} p_k = 1$ , we have

$$p_0 = \left( \sum_{k=0}^{\infty} r_k \right)^{-1} \quad (4.9)$$

We remark that steady state cannot be reached if  $\lambda_k, \mu_k$  are s.t.  $\sum_{k=0}^{\infty} r_k = \infty$ . If  $\lambda, \mu$  and  $\rho = \frac{\lambda}{s\mu}$  are as in section 4.1, then steady state can be reached if  $\rho < 1$ .

If  $s, \bar{L}, \bar{L}_q$  as defined in section 4.1, and  $\bar{\lambda}$  is as in 4.2, then

$$\bar{L} = \sum_{k=0}^{\infty} k p_k, \quad \bar{L}_q = \sum_{k=s}^{\infty} (k-s) p_k, \quad \text{and} \quad \bar{\lambda} = \sum_{k=0}^{\infty} \lambda_k p_k \quad (4.10)$$

Now, consider the M/M/1 system. For  $\lambda, \mu, \rho$  as above, with  $\rho = \frac{\lambda}{\mu} < 1$ , we have

$$4.8 \Rightarrow r_k = \left( \frac{\lambda}{\mu} \right)^k = \rho^k, \quad \forall k \geq 0.$$

$$4.9 \Rightarrow p_0 = \left( \sum_{k=0}^{\infty} r_k \right)^{-1} = \left( \sum_{k=0}^{\infty} \rho^k \right)^{-1} = 1 - \rho.$$

$$4.10 \Rightarrow \bar{L} = \sum_{k=0}^{\infty} k p_k = \sum_{k=0}^{\infty} k r_k p_0 = (1 - \rho) \sum_{k=0}^{\infty} k \rho^k = (1 - \rho) \rho \frac{d}{d\rho} \sum_{k=0}^{\infty} \rho^k = (1 - \rho) \rho \frac{d}{d\rho} \frac{1}{1 - \rho} = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}$$

Similarly, 4.10  $\Rightarrow$

$$\bar{L}_q = \sum_{k=1}^{\infty} (k-1)p_k = \bar{L} - (1-p_0) = \frac{\lambda^2}{\mu(\mu-\lambda)}$$

Note that if  $\rho = \frac{\lambda}{\mu} > 1$ , then  $\sum_{k=0}^{\infty} r_k = \sum_{k=0}^{\infty} \rho^k = \infty$ , and we cannot have steady state.

Back to  $\rho < 1$ . Let  $W$  be a sojourn time RV. If the customer  $C$  finds  $k$  customers in the queue, then  $C$  has to wait through  $k+1$  service times, including  $C$ 's own, but not the customer currently in service (see Prop 4.3). If  $X_1, \dots, X_{k+1}$  are the RVs of the service times, each exponential with parameter  $\mu$ , then letting  $S_{k+1} = X_1 + \dots + X_{k+1}$ ,  $k \geq 0$ , we have

$$\mathbb{P}(W > t) = \sum_{k=0}^{\infty} p_k \mathbb{P}(S_{k+1} > t)$$

It can be shown from this that

$$\mathbb{P}(W > t) = e^{-\mu(1-\rho)t}, \quad t \geq 0$$

Thus,  $W$  has the exponential distribution with parameter  $\mu(1-\rho)$ , and

$$\mathbb{E}W = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu-\lambda}$$

Similarly, let  $W_q$  be a queueing time RV. If  $C$  finds no customers already in the system, then  $\mathbb{P}(W_q = 0) = 1 - \rho$ . If  $c$  finds  $k \geq 1$  customers in the system, then  $C$  has to wait through  $k$  service times, so

$$\begin{aligned} \mathbb{P}(W_q > t) &= \sum_{k=1}^{\infty} p_k \mathbb{P}(S_k > t) = \sum_{k=1}^{\infty} (1-\rho)\rho^k \mathbb{P}(S_k > t) \\ &= \rho \sum_{k=0}^{\infty} p_k \mathbb{P}(S_{k+1} > t) = \rho \mathbb{P}(W > t) \\ &= \rho e^{-\mu(1-\rho)t}, \quad t \geq 0 \end{aligned}$$

And  $\mathbb{E}W_q = \mathbb{E}W - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu-\lambda)}$ .