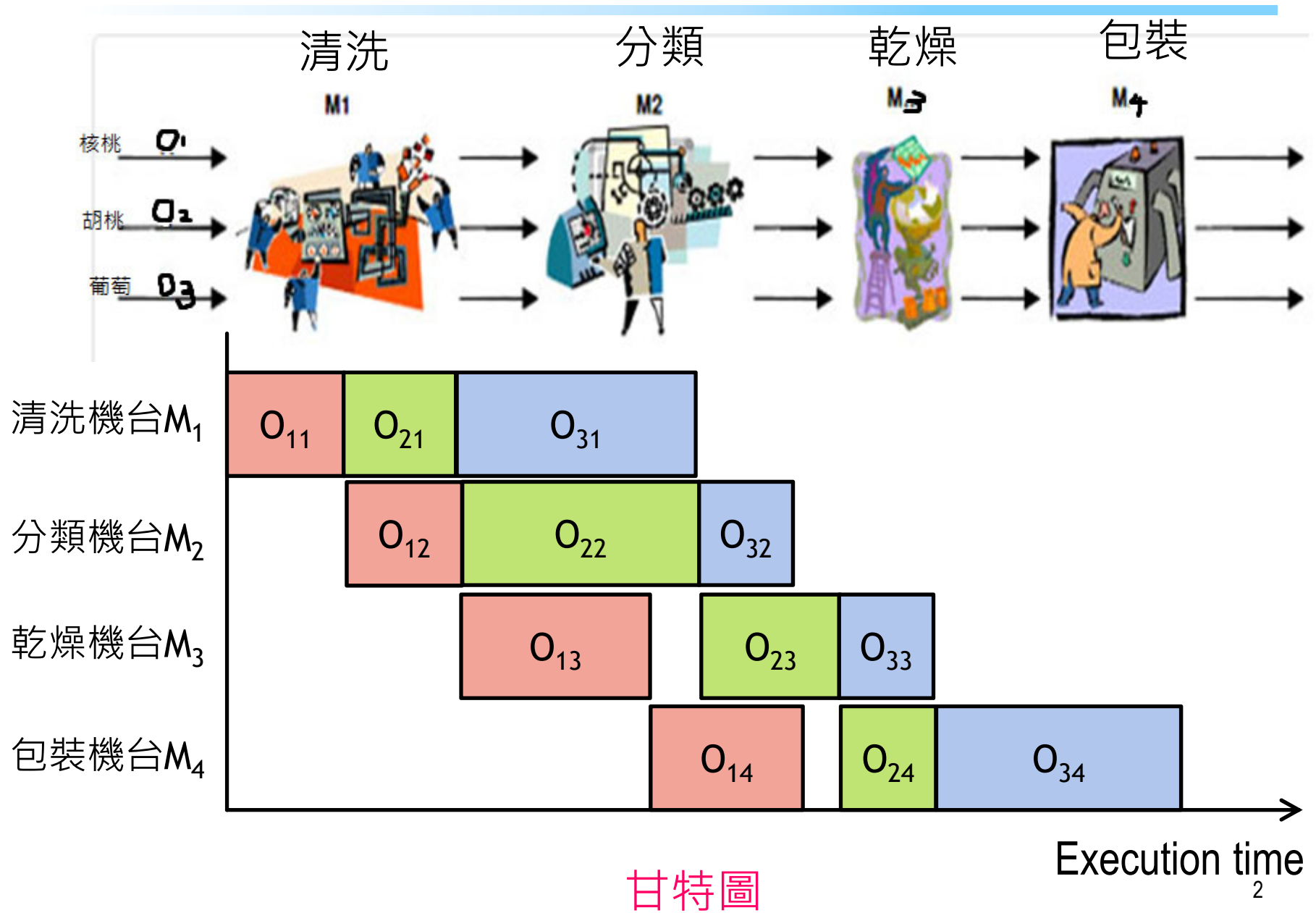


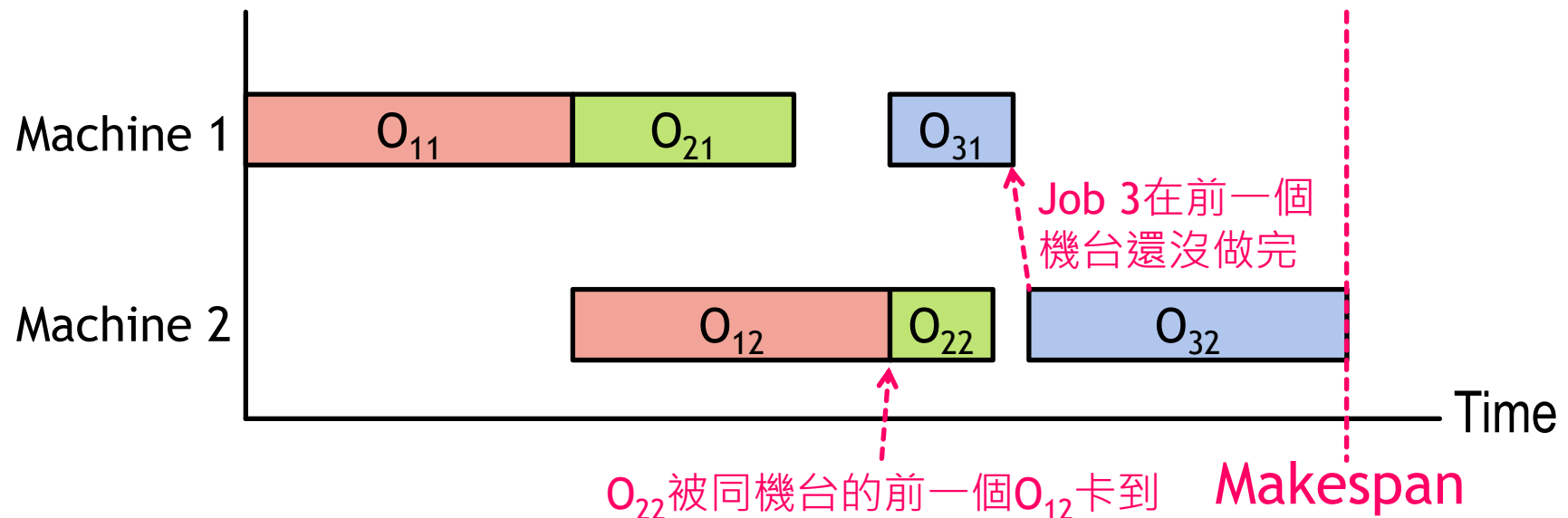
# 流程型生產排程問題 (Flowshop scheduling Problem)

# Example



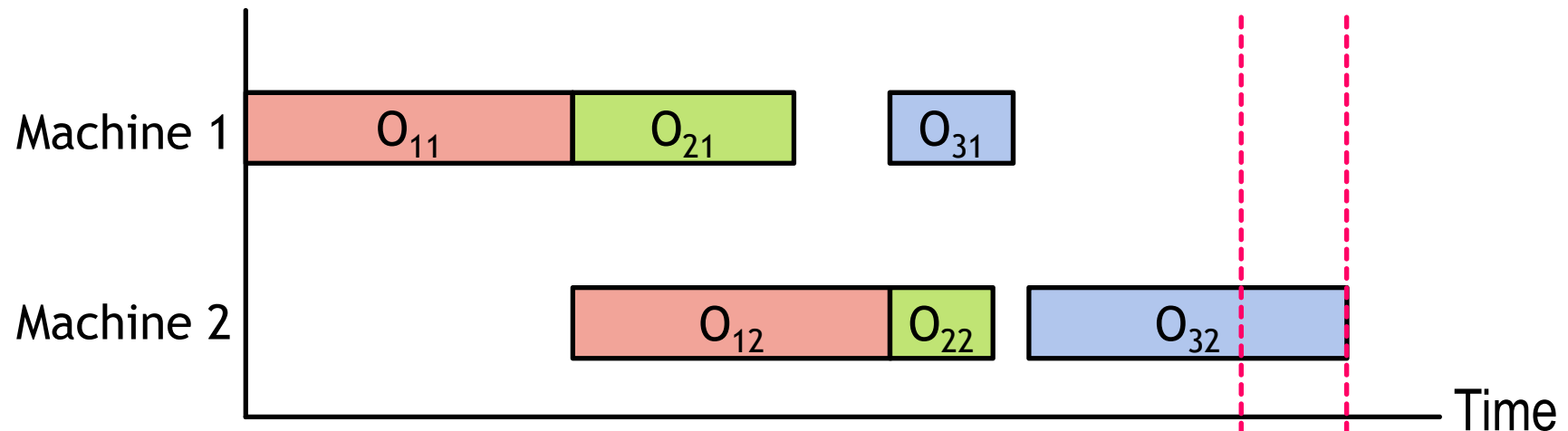
# Flow Shop Scheduling Problem

- The **processing order** of the **jobs** on the resources is **the same** for each subsequent step of processing
- Proved to be **NP-hard problem**
- Objective is to **minimize** the **MAKESPAN** (總完工時間)
  - ✓  $O_{ij}$  = Operation of job  $i$  at machine  $j$
  - ✓  $P_{ij}$  = Processing time of operation  $O_{ij}$

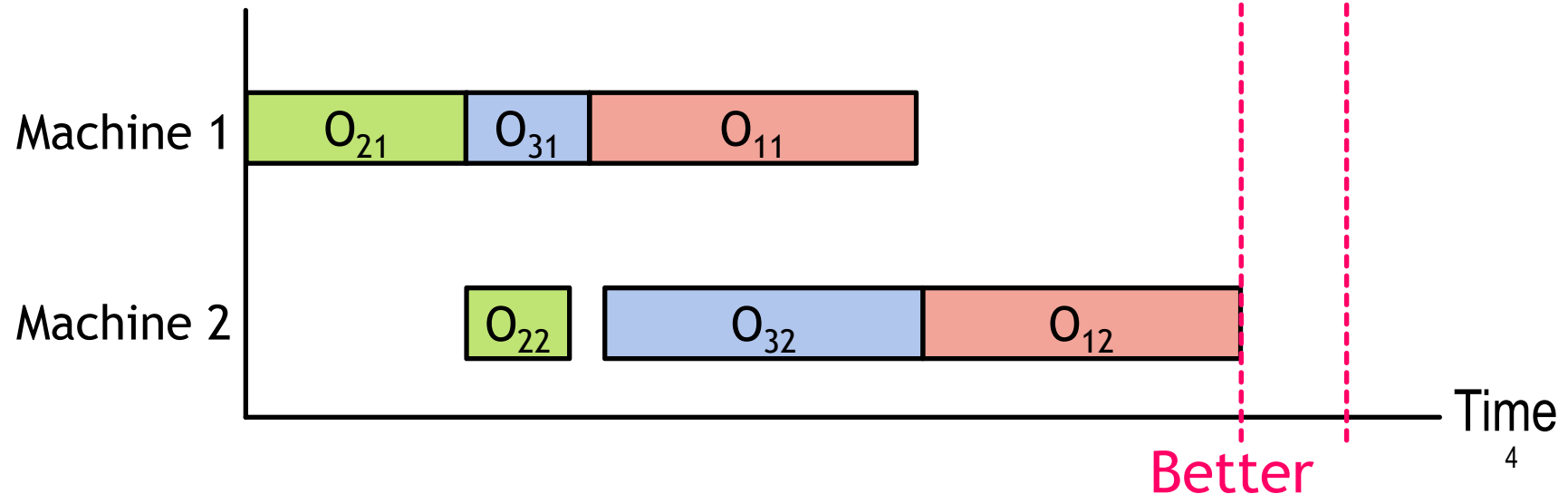


# Comparison of 2 solutions

- Job 1 → Job 2 → Job 3



- Job 2 → Job 3 → Job 1



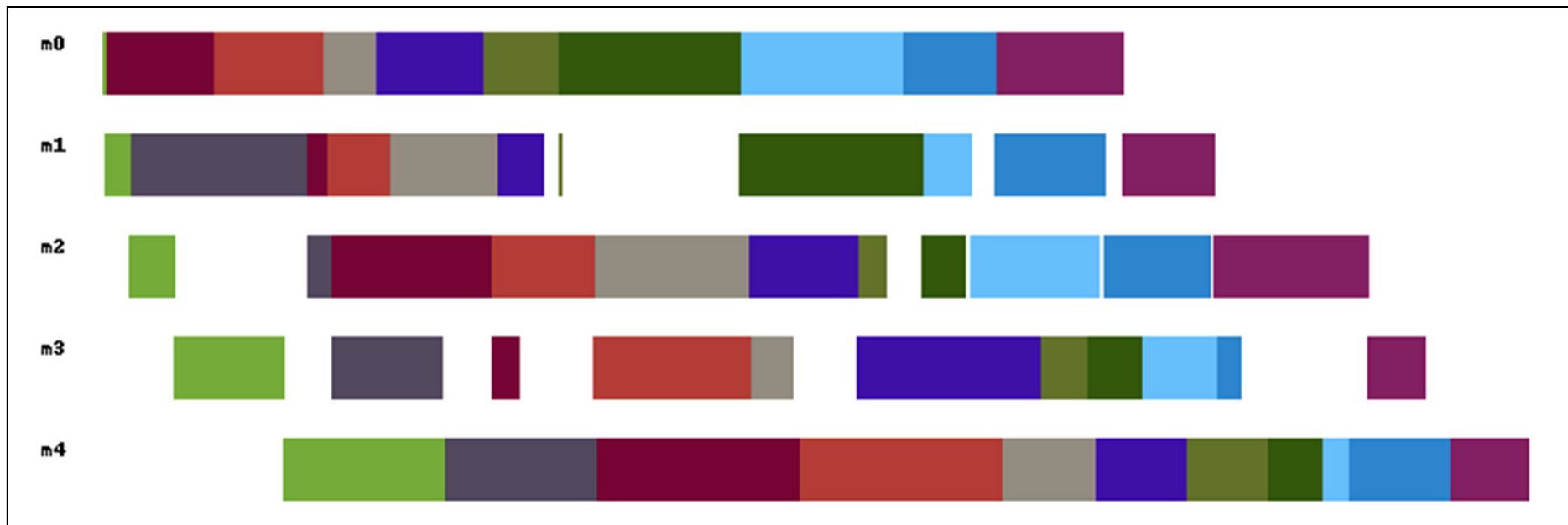
# 解的編碼

- 解的編碼

- 工件的所有編號的排列

- 問題實例 (5個機台，11個job)

- 解為11個 job 的排列：[7, 2, 4, 10, 8, 6, 0, 9, 5, 3, 1]



# Python code for solution representation

- 解的編碼
  - 工件的所有編號的排列
- 問題實例 (5個機台，11個job)
  - 解為11個 job 的排列：[7, 2, 4, 10, 8, 6, 0, 9, 5, 3, 1]

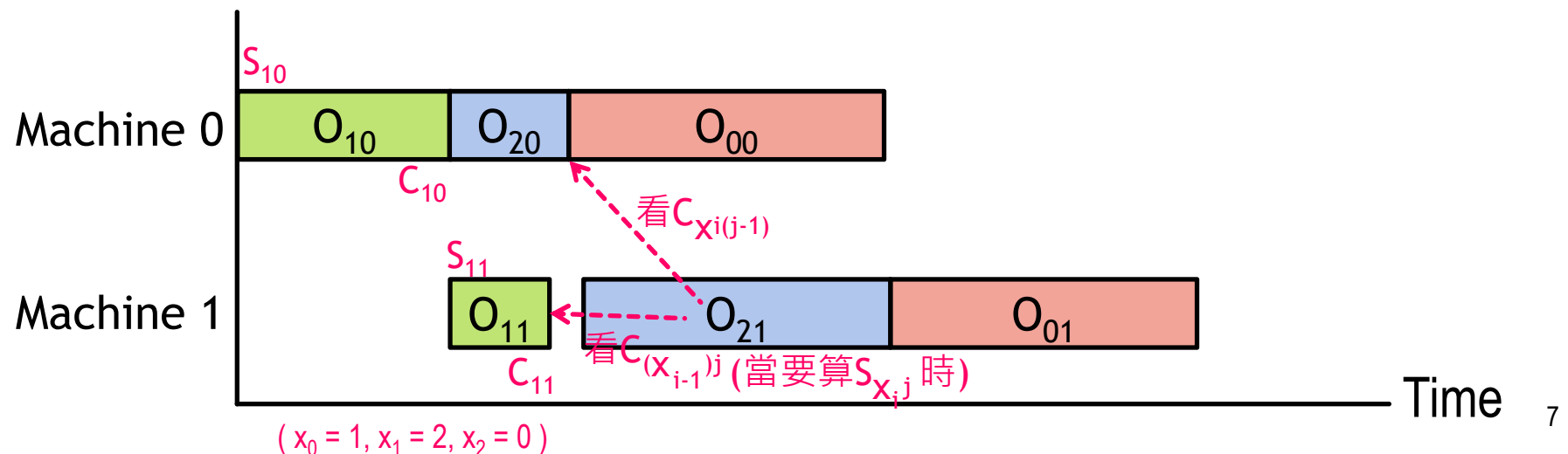
```
44 def initPop():                # 初始化群體
45     p = []
46
47     for i in range(NUM_CHROME):
48         # 產生 0, 1, ..., NUM_BIT-1 的排列
49         p.append(np.random.permutation(NUM_BIT))
50
51     return p
```

# 解的解碼

- 給定一個編碼  $x_0, x_1, \dots, x_{n-1}$
- 解的解碼演算法：

$S_{ij}$  表示 job  $i$  在 machine  $j$  的開始時間  
 $P_{ij}$  表示 job  $i$  在 machine  $j$  的加工時間  
 $C_{ij}$  表示 job  $i$  在 machine  $j$  的完成時間

1. 初始化 job  $x_0$  的所有開始時間  $S_{x_0j}$  與結束時間  $C_{x_0j}$
2. 考慮 jobs  $x_1, x_2, \dots, x_{n-1}$ 
  1.  $S_{x_ij} = \max\{C_{(x_{i-1})j}, C_{x_i(j-1)}\}$
  2.  $C_{x_ij} = S_{x_ij} + P_{x_ij}$
3. 考慮 job  $x_{n-1}$  的完工時間即為 makespan



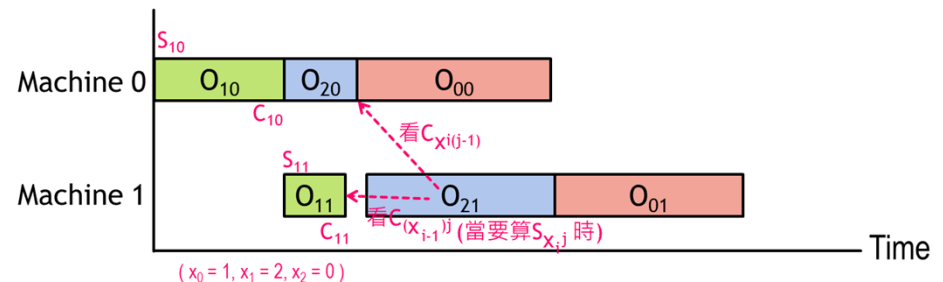
# Exercise

- 在“GA10-flowshop-ex.py”寫上fitFunc函數的內容

- $S[i][j]$  表示 job  $i$  在 machine  $j$  的開始時間
- $C[i][j]$  表示 job  $i$  在 machine  $j$  的完成時間
- $pTime[i][j]$  表示 job  $i$  在 machine  $j$  的加工時間

- 演算法

- ✓ (初始化) 計算第 $X[0]$ 個 job 在各 machine 的開始時間  $S[x[0]][j]$
- ✓ (初始化) 計算上述的完成時間  $C[x[0]][j]$
- ✓ 依序考慮  $x[1], x[2], \dots, x[i], \dots$ 
  - ▶ 計算第 $X[i]$ 個 job 在各 machine 的開始時間  $S[x[i]][j]$  (公式在前一頁)
  - ▶ 計算上述的完成時間  $C[x[i]][j]$  (公式在前一頁)



- ✓  $C[X[NUM\_JOB - 1]][NUM\_MACHINE - 1]$  即為最大完工時間