Modèles et algorithmes en ordonnancement

Ordonnancement d'atelier : flowshop à deux machines

Safia Kedad-Sidhoum

CNAM
safia.kedad_sidhoum@cnam.fr

ORDO - M2 MPRO, 2025-2026

Introduction

- Ordonnancement d'atelier: jobshop, flowshop, openshop
- Machines organisées en série (étages)
- Ordonnancement flowshop:
  - m machines spécifiques: $M_1, \cdots, M_m$
  - n tâches: $J_1, \cdots, J_n$
  - Une tâche est considérée comme une suite d'opérations O_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, n$:
    - O_{ij} est exécutée sur la machine M_i
    - Pour tout $i = 1, \cdots, m-1$, O_{ij} doit être exécutée avant $O_{i+1,j}$
    - La durée de l'opération O_{ij} est p_{ij}
  - On s'intéresse à la minimisation de la durée totale de l'ordonnancement (C_{max})

Introduction

- Ordonnancement d'atelier: jobshop, flowshop, openshop
- Machines organisées en série (étages)
- Ordonnancement flowshop:
  - m machines spécifiques: $M_1, \cdots, M_m$
  - n tâches: $J_1, \cdots, J_n$
  - Une tâche est considérée comme une suite d'opérations O_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, n$:
    - O_{ij} est exécutée sur la machine M_i
    - Pour tout $i = 1, \cdots, m-1$, O_{ij} doit être exécutée avant $O_{i+1,j}$
    - La durée de l'opération O_{ij} est p_{ij}
  - On s'intéresse à la minimisation de la durée totale de l'ordonnancement (C_{max})

Safia Kedad-Sidhoum - 2 (CNAM) Ordonnancement 2025-2026 2 / 7

Introduction

- Ordonnancement d'atelier: jobshop, flowshop, openshop
- Machines organisées en série (étages)
- Ordonnancement flowshop:
  - m machines spécifiques: $M_1, \cdots, M_m$
  - n tâches: $J_1, \cdots, J_n$
  - Une tâche est considérée comme une suite d'opérations O_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, n$:
    - O_{ij} est exécutée sur la machine M_i
    - Pour tout $i = 1, \cdots, m-1$, O_{ij} doit être exécutée avant $O_{i+1,j}$
    - La durée de l'opération O_{ij} est p_{ij}
  - On s'intéresse à la minimisation de la durée totale de l'ordonnancement (C_{max})

Safia Kedad-Sidhoum - 2 (CNAM) Ordonnancement 2025-2026 2 / 7

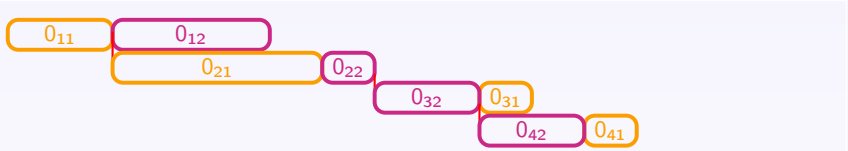
Ordonnancement de permutation

- L'ordre de passage des tâches sur une machine M_k est défini par $\Pi(k)$
- Une solution est dite de permutation si $\Pi(1) = \Pi(2) = \cdots \Pi(m)$.
- Il existe donc $n!$ solutions de permutation.

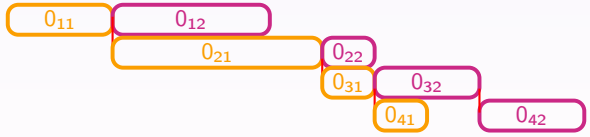
Exemple

$n = 2$ et $m = 4$

- Flowshop



- Flowshop de permutation (ordre de passage des tâches identique sur les quatre machines)



Propriété de dominance

- Les solutions telles que $\Pi(1) = \Pi(2)$ et $\Pi(m-1) = \Pi(m)$ sont dominantes.
- Pour $m \leq 3$, les solutions de permutation sont donc dominantes.

Complexité des problèmes de flowshop

- $Fm||C_{max}$ pour $m \geq 3$ est NP-difficile au sens fort.
- $F2||C_{max}$ est polynomial: Algorithme de Johnson.

Algorithme de Johnson ($F2||C_{max}$)

1. $L_g \leftarrow ()$, $L_d \leftarrow ()$
2. Pour $k = 1, \cdots, n$:
   Déterminer l'opération O_{ij} de durée minimale des tâches non encore ordonnancés.
   - Si $i = 1$ alors $L_g \leftarrow L_g.(J_j)$
   - Si $i = 2$ alors $L_d \leftarrow (J_j).L_d$
3. Retourner (L_g, L_d)
   (ordre de passage optimal des tâches sur les machines)

Exemple (au tableau)
$n = 5$, $p_1 = (3, 5, 1, 6, 7)$ et $p_2 = (6, 2, 2, 6, 5)$.