MARO 018 - Optimization & Operational Research

Arnaud Vandaele

Faculté Polytechnique
Service MAthématique et Recherche Opérationnelle

UMONS

Année académique 2024-2025

Séance 2 :
Ordonnacement de production

Séance 2 – Ordonnacement de production

Tables des matières

Plan

Problèmes d'ordonnancement d'ateliers

Les problèmes *d'ordonnacement d'ateliers* sont une catégorie importante de problèmes d'optimisation combinatoire.

De façon générale, les données de ces problèmes sont

- une liste de n jobs (tâches), et
- une liste de m machines.

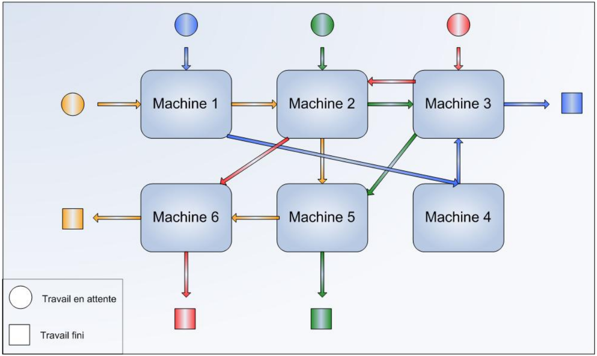
Le but du problème consiste à optimiser l'affectation des tâches aux machines afin de minimiser une certaine fonction objective.

Il existe de nombreux problèmes différents de programmation optimale des tâches, qui diffèrent par la nature des tâches, la nature des machines, les restrictions sur la programmation et la fonction objective.

Exemple 1 : le problème de Job-shop

Etant données $m = 6$ machines et $n = 4$ jobs :

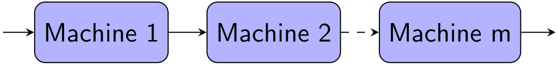
- Le job orange doit passer par M1, puis M2, puis M5, puis M6,
- Le job bleu doit passer par M1, puis M4, puis M3,
- Le job vert doit passer par M2, puis M3, puis M5,
- Le job rouge doit passer par M3, puis M2, puis M6,



comment planifier ces tâches afin de minimiser le temps total d'exécution ?

Exemple 2 : le problème de flowshop

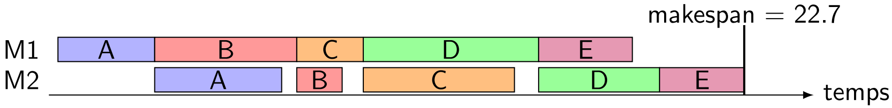
Dans ce cas particulier, les chemins des tâches sur les machines sont les mêmes :



Exemple avec $m = 2$ machines et $n = 5$ jobs, et les durées de traitement :

	Job A	Job B	Job C	Job D	Job E
Machine 1	3.2	4.7	2.2	5.8	3.1
Machine 2	4.2	1.5	5	4	2.8

Les jobs doivent d'abord être exécutés sur M1 puis sur M2, mais dans quel ordre ?



Plan

Problème 1 : minimisation de C_{mean} dans le cas $m = 1$

Déterminer l'ordonnacement des 7 tâches suivantes sur 1 machine afin de minimiser le temps moyen de réalisation.

Jobs	1	2	3	4	5	6	7
durées d_i	6	4	6	3	2	7	1

(+ preuve)

Problème 2 : minimisation de $L_{\max}$ dans le cas $m = 1$

Déterminer l'ordonnacement des 6 tâches suivantes sur 1 machine afin de minimiser le retard maximal.

Jobs	1	2	3	4	5	6
durées d_i	1	1	2	4	1	3
échéances e_i	7	3	8	12	9	3

(+ preuve)

Problème 3 : minimisation de N_t dans le cas $m = 1$

Déterminer l'ordonnacement des 6 tâches suivantes sur 1 machine afin de minimiser le nombre de jobs en retard.

Jobs	1	2	3	4	5	6
durées d_i	10	3	4	8	10	6
échéances e_i	15	6	9	23	20	30

Problème 4 : minimisation de $C_{\max}$ dans le cas $m = 2$

Déterminer l'ordonnacement des 5 tâches suivantes sur 1 machine afin de minimiser le temps de réalisation total.

	Job A	Job B	Job C	Job D	Job E
Machine 1	3.2	4.7	2.2	5.8	3.1
Machine 2	4.2	1.5	5	4	2.8

Problème 5 : min. $C_{\max}$ dans un cas particulier avec $m = 3$

Déterminer l'ordonnacement des 6 tâches suivantes sur 1 machine afin de minimiser le temps de réalisation total.

	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6
Machine 1	4	6	3	5	8	4
Machine 2	1	2	1	3	2	1
Machine 3	3	9	2	7	6	1

Plan

Plan

Exercice : vers la complexité

Tentez de déterminer la meilleure permutation minimisant $C_{\max}$ pour cet exemple avec $n = 9$ jobs sur $m = 3$ machines :

	Job C_0	Job C_1	Job C_2	Job E_1	Job E_2	Job E_3	Job E_4	Job E_5	Job E_6
Mach. 1	0	30	30	0	0	0	0	0	0
Mach. 2	15	15	15	4	5	5	5	5	6
Mach. 3	30	30	0	0	0	0	0	0	0

Tenter de déterminer la solution optimale.

Reduction

3-PARTITION est un problème NP-hard :

INSTANCE: Un ensemble d'entiers $A = \{a_1, ..., a_{3n}\}$ tel que $\sum_i a_i = nB$ et $\frac{B}{4} < a_i < \frac{B}{2}$.
QUESTION: est-il possible de trouver une partition des éléments de A en triplets telle que la somme de chaque triplet soit B.

Exemple avec $n = 2$ et $B = 15$: $A = \{4, 5, 5, 5, 5, 6\}$
Réponse : OUI avec $\{(4, 5, 6), (5, 5, 5)\}$.

Que peut-on dire sur la complexité du problème de PERMUTATION FLOWSHOP ?

INSTANCE: n jobs et leur durée respective sur M_1, M_2 et M_3 et un naturel D.
QUESTION: est-il possible de trouver une permutation des n jobs afin que $C_{\max} \leq D$?

- A partir de n'importe quelle instance de 3-PARTITION, on décrit une procédure qui va créer une instance de PERMUTATION FLOWSHOP en temps polynomial.
- Ensuite, il faut prouver :
  (instance-OUI $\Rightarrow$ instance-OUI) et (instance-OUI $\Leftarrow$ instance-OUI).