

0.1 Introduction générale

Ce cours présente les modèles et les algorithmes essentiels utilisés pour résoudre des problèmes de planification et d'ordonnancement, en s'appuyant sur des techniques de recherche opérationnelle.

L'ordonnancement est au coeur du pilotage de la production. Il consiste à organiser dans le temps l'exécution de tâches sur des ressources limitées, tout en respectant des contraintes temporelles, de précedence et de ressources, et en optimisant un ou plusieurs critères de performance.

0.2 Formulation des problèmes d'ordonnancement

0.2.1 Définition et importance

L'**ordonnancement** est le processus de répartition des tâches, ou opérations, sur des ressources, généralement des machines, au cours du temps, dans le but de satisfaire des contraintes et d'optimiser une fonction objectif.

Son importance en planification de la production réside dans :

- l'amélioration de l'utilisation des ressources : machines et main-d'oeuvre ;
- la réduction des temps d'attente et des en-cours ;
- la garantie du respect des délais de livraison ;
- la minimisation des coûts liés aux retards et aux stocks.

0.2.2 Concepts de base

- **Tâche** : opération élémentaire nécessitant un temps d'exécution, notée i ou j .
- **Machine** : ressource sur laquelle s'exécutent les tâches, notée M ou m .
- **Durée d'exécution** p_{ij} ou p_i : temps nécessaire pour réaliser la tâche i sur la machine j .
- **Date de disponibilité** r_i : moment à partir duquel la tâche i est prête à être exécutée.
- **Date d'échéance** d_i : date limite souhaitée pour la fin de la tâche i .
- **Date de fin** C_i : instant où la tâche i se termine.

0.2.3 Contraintes temporelles et de ressources

- **Contraintes de ressources** : une machine ne peut exécuter qu'une seule tâche à la fois ; une tâche ne peut pas être exécutée simultanément sur plusieurs machines.
- **Contraintes de précedence** : certaines tâches doivent être terminées avant que d'autres ne commencent, conformément aux gammes opératoires.
- **Contraintes temporelles** : une tâche ne peut pas commencer avant sa date de disponibilité r_i , et il faut idéalement la terminer avant sa date d'échéance d_i .

0.2.4 Fonctions objectifs

Les critères les plus couramment optimisés sont :

- **Makespan** $C_{\max}$: date de fin de la dernière tâche,

$$C_{\max} = \max_i C_i.$$

Minimiser $C_{\max}$ revient à minimiser le temps total de production.

- **Somme des dates de fin** $\sum C_i$: permet de réduire le temps moyen de séjour des tâches dans l'atelier.

- **Retard maximal** $L_{\max}$:

$$L_{\max} = \max_i (C_i - d_i).$$

Si $L_{\max} \leq 0$, alors tous les délais sont respectés.

- **Somme des retards** $\sum T_i$: où

$$T_i = \max(0, C_i - d_i)$$

est le retard effectif de la tâche i .

0.2.5 Formulation mathématique générale

Soit n tâches et m machines.

Variable de décision :

$$x_{ijk} = \begin{cases} 1, & \text{si la tâche } i \text{ est la } k\text{-ième exécutée sur la machine } j, \\ 0, & \text{sinon.} \end{cases}$$

Variable continue : $C_{\max}$, le makespan.

Fonction objectif :

$$\min C_{\max}.$$

Sous contraintes :

$$\sum_{k=1}^n x_{ijk} = 1, \quad \forall i, j, \quad (1)$$

$$\sum_{i=1}^n x_{ijk} \leq 1, \quad \forall j, k, \quad (2)$$

$$C_i \geq C_{i-1} + p_{ij}, \quad (3)$$

$$C_{\max} \geq C_i, \quad \forall i. \quad (4)$$

La première contrainte impose que chaque tâche passe une fois sur chaque machine. La deuxième garantit qu'une seule tâche occupe une position donnée sur une machine. La troisième traduit les relations de précédence, et la dernière définit le makespan.

0.3 Ordonnancement sur une seule machine

0.3.1 Définition et hypothèses

- n tâches sont disponibles à $t = 0$, c'est-à-dire $r_i = 0$.
- Une seule machine est disponible.
- Il n'y a pas de préemption : une tâche s'exécute jusqu'à son terme.
- L'objectif est de déterminer la séquence optimale selon un critère donné.

0.3.2 Règles de priorité classiques

- **PAPS** (*First Come, First Served*) : les tâches sont ordonnancées selon leur ordre d'arrivée.
- **SPT** (*Shortest Processing Time; durée opératoire la plus courte*) : tri des tâches par durée p_i croissante. Cette règle est optimale pour minimiser $\sum C_i$.
- **EDD** (*Earliest Due Date; date de livraison la plus rapprochée*) : tri des tâches par échéance d_i croissante. Cette règle est optimale pour minimiser $L_{\max}$.
- **LPT** (*Longest Processing Time; durée opératoire la plus longue*) : tri des tâches par durée p_i décroissante.

0.3.3 Algorithme SPT

Étapes

Entrée : une liste de tâches avec leurs durées p_i .

1 Trier les tâches par ordre croissant de durée p_i . La tâche la plus courte passe en premier.

1 Initialiser la date de fin de la tâche 0 (début de l'ordonnancement) : $C_0 = 0$.

1 Parcourir les tâches dans l'ordre trié, de la première à la dernière. Pour chaque tâche i :

- Calculer sa date de fin : $C_i = C_{i-1} + p_i$. Elle commence quand la tâche précédente se termine et dure p_i .
- Enregistrer la valeur C_i .

1 Retourner la séquence ordonnée des tâches et leurs dates de fin C_i .

0.3.4 Exemple numérique résolu

Soit quatre tâches à exécuter sur une seule machine.

Tâche	Durée d'opération p_i	Échéance d_i ; date de livraison promise
A	4	10
B	2	6
C	6	12
D	1	5

Application de la règle SPT :

1. Tri par durée croissante :

$$D(1), B(2), A(4), C(6).$$

2. Calcul des dates de fin :

- $C_D = 1$;
- $C_B = 1 + 2 = 3$;
- $C_A = 3 + 4 = 7$;
- $C_C = 7 + 6 = 13$.

3. Calcul des retards $L_i = C_i - d_i$:

- $L_D = 1 - 5 = -4$, donc pas de retard ;
- $L_B = 3 - 6 = -3$;
- $L_A = 7 - 10 = -3$;
- $L_C = 13 - 12 = 1$, donc un retard de 1.

4. Indicateurs :

$$\sum C_i = 1 + 3 + 7 + 13 = 24, \quad L_{\max} = 1.$$

Exemple numérique résolu

Soit quatre tâches à exécuter sur une seule machine.

Tâche	Durée d'opération p_i	Échéance d_i ; date de livraison promise
A	4	5
B	2	9
C	6	14
D	1	12

Règle SPT (durée p_i croissante). Séquence : D, B, A, C.

Tâche	D	B	A	C
p_i	1	2	4	6
C_i	1	3	7	13
d_i	12	9	5	14
$L_i = C_i - d_i$	-11	-6	2	-1

$\sum C_i = 24$, $L_{\max} = 2$, 1 tâche en retard (A).

Règle PAPS / FCFS (ordre d'arrivée, ici A, B, C, D). Séquence : A, B, C, D.

Tâche	Durée d'opération p_i	Échéance d_i ; date de livraison promise
A	4	10
B	2	6
C	6	12
D	1	5

$\sum C_i = 35$, $L_{\max} = 1$, 1 tâche en retard (D).

Règle EDD (échéance d_i croissante). Séquence : A, B, D, C.

Tâche	A	B	D	C
p_i	4	2	1	6
C_i	4	6	7	13
d_i	5	9	12	14
$L_i = C_i - d_i$	-1	-3	-5	-1

$\sum C_i = 30$, $L_{\max} = -1$, aucune tâche en retard.

Règle LPT (durée p_i décroissante). Séquence : C, A, B, D.

Tâche	C	A	B	D
p_i	6	4	2	1
C_i	6	10	12	13
d_i	14	5	9	12
$L_i = C_i - d_i$	-8	5	3	1

$\sum C_i = 41$, $L_{\max} = 5$, 3 tâches en retard (A, B, D).

Comparaison.

Règle	Séquence	$\sum C_i$	$L_{\max}$	Tâches en retard
SPT	D-B-A-C	24	2	1
PAPS	A-B-C-D	35	1	1
EDD	A-B-D-C	30	-1	0
LPT	C-A-B-D	41	5	3

Conclusion. SPT minimise la somme des dates de fin $\sum C_i$ (24), et EDD minimise $L_{\max}$ (-1). Les deux critères ne sont pas atteints par la même règle. LPT est la pire ici sur les deux critères, et PAPS ne tient compte ni des durées ni des échéances.