

# Chapitre 2 : L'ordonnancement de la production

VIII

Ordonnancement	15
Formulation d'un problème d'ordonnancement	16
Typologie des problèmes d'ordonnancements	19
Formalisation des problèmes d'ordonnancements	21
Exercice	24
La théorie de la complexité	24

La théorie d'ordonnancement est une branche de la recherche opérationnelle, qui consiste à ordonner un ensemble d'opérations tout en satisfaisant un ensemble de contraintes et en optimisant un ou plusieurs objectifs. L'ordonnancement joue un rôle essentiel dans de nombreux secteurs d'activités à savoir, en : informatique (ordonnancement de processus), administration (établissement d'un emploi du temps, gestion des ressources humaines), industrie (gestion des ateliers de production), construction (gestion des chantiers routiers), logistique (gestion des livraisons et des stocks).

Parmi ces nombreux types de problèmes d'ordonnancement, nous nous sommes intéressés dans le cadre de cette thèse aux problèmes d'ordonnancement d'ateliers dans les systèmes de production.

Un atelier de production est un espace physique où la fabrication se déroule. Il est composé de ressources humaines et matérielles, et caractérisé par les types de tâches à exécuter, les types de ressources et la gamme de fabrication, que nous présentons en détail dans les sous-sections suivantes.

## 1. Ordonnancement

Nombreuses sont les définitions proposées au problème d'ordonnancement d'atelier, nous tirons de la littérature les trois définitions ci-dessous:

### Définition

Définition 1 : Scheduling is the process of organizing, choosing, and timing resource usage to carry out

all the activities necessary to produce the desired outputs at the desired times, while satisfying a large number of time and relationship constraints among the activities and the resources.

### Définition

Définition 2 : Scheduling is concerned with the allocation of scarce resources to activities with the objective of optimizing one or more performance measure.

### Définition

Définition 3 : Scheduling is a decision-making process that is used on a regular basis in many manufacturing and services industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives.

D'après les définitions ci-dessus, on retrouve l'aspect commun de l'affectation de ressources aux tâches. Donc nous pouvons dire que l'ordonnement d'ateliers consiste à programmer dans le temps l'exécution des tâches selon la disponibilité des ressources pour répondre à un ou plusieurs d'objectifs, tout en respectant les contraintes techniques de fabrication.

## 2. Formulation d'un problème d'ordonnement

Les tâches	16
Les ressources	17
Les contraintes	18
Les objectifs	18

Dans un problème d'ordonnement, quatre notions fondamentales interviennent : les tâches, les ressources, les contraintes et les objectifs. Dans ce qui suit, on donnera une définition détaillée de chacun de ces notions.

### 2.1. Les tâches

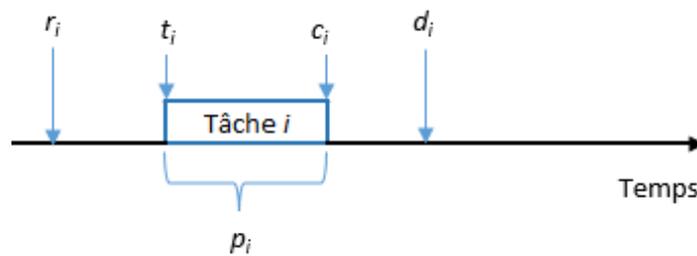
Une tâche est une entité élémentaire de travail localisé dans le temps par une date de début  $t_i$  (start time) et une date de fin  $c_i$  (completion time), dont la réalisation est caractérisée par une

durée positive  $p_i$  (processing time) telle que  $p_i = c_i - t_i$

Certaines caractéristiques relatives à l'exécution d'une tâche sont définies ainsi :

- Une date de disponibilité  $r_i$  (release date) qui correspond à la date de début au plus tôt.
- Une date d'échéance  $d_i$  (due date) qui correspond à la date de fin au plus tard.
- Un poids  $w_i$  (weight) qui représente le facteur de priorité qui dénote l'importance de la tâche i relativement aux autres.

La figure suivante donne une représentation de la tâche en désignant ses principales caractéristiques.



Caractéristique d'une tâche

On distingue deux types de tâches :

- Des tâches morcelables (préemptibles) qui peuvent être exécutées par morceaux par une ou plusieurs ressources.
- Des tâches non-morcelables (indivisibles) qui doivent être exécutées en une seule fois et ne peuvent pas être interrompues avant qu'elles ne soient complètement achevées.

### Remarque

Généralement, en ordonnancement d'atelier, le terme « tâche » correspond à une « opération » et le terme « travail » ou « job » désigne l'ensemble d'opérations constituant le même job.

## 2.2. Les ressources

1. Selon leurs disponibilités au cours du temps, on trouve :

- Les ressources renouvelables, comme c'est le cas pour les machines, personnels, équipements, etc. La ressource est dite renouvelable si après avoir été utilisée par une ou plusieurs opérations, elle est à nouveau disponible en même quantité.
- Les ressources non-renouvelables, souvent appelées ressources consommables ou bien ressources financières. On dit que la ressource est non-renouvelable si sa disponibilité décroît après avoir été allouée à une opération. C'est le cas pour la matière première, budget.
- Les ressources doublement contraintes, ces ressources combinent les contraintes liées aux deux catégories précédentes. Leur utilisation instantanée et leur consommation globale sont toutes les deux limitées. C'est le cas des ressources d'énergie (pétrole, électricité, etc.).

2. Selon leurs capacités, on trouve :

- Les ressources disjonctives (ou non-partageables), il s'agit des ressources qui ne peuvent exécuter qu'une seule opération à la fois c'est le cas par exemple de machine-outil ou robot manipulateur.

- Les ressources cumulatives (ou partageables), il s'agit des ressources qui peuvent être utilisées par plusieurs opérations simultanément (équipes d'ouvriers, poste de travail, etc.).

## 2.3. Les contraintes

Les contraintes expriment des restrictions sur les valeurs que peuvent prendre conjointement les variables de décision. En d'autres termes, les contraintes représentent les conditions à respecter lors de la construction de l'ordonnancement pour qu'il soit réalisable. Plus les contraintes sont nombreuses, plus le problème d'ordonnancement devient plus difficile.

Dans les problèmes d'ordonnancement, deux types de contraintes sont distingués : les contraintes temporelles et les contraintes de ressources.

1. Les contraintes temporelles décrivent les interdépendances temporelles entre les opérations, elles intègrent :

- Les contraintes de dates limites : c'est des contraintes imposées individuellement à chaque opération  $i$ . Par exemple, l'opération  $i$  ne peut débuter avant une certaine date (livraison de matière première, conditions climatiques, etc.) ou encore l'opération  $i$  ne peut commencer avant sa date de disponibilité  $r_i$  et doit être terminée avant une date d'échéance  $d_i$ .
- Les contraintes d'antériorité : c'est des contraintes qui relient la date de début ou la date de fin de deux opérations par une relation linéaire. De manière générale, c'est des contraintes qui décrivent le positionnement relatif de certaines opérations par rapport à d'autres, c'est le cas par exemple des gammes opératoires dans les d'ateliers de production.

2. Les contraintes de ressources traduisent l'utilisation et la disponibilité des ressources utilisées par les opérations. Deux types de contraintes liées à la nature cumulative ou disjonctive des ressources peuvent alors être distingués.

- Les contraintes disjonctives : ces contraintes imposent la non-réalisation simultanée de deux opérations sur la même ressource.
- Les contraintes cumulatives : ces contraintes expriment le fait qu'à tout instant, le total des ressources utilisées ne dépasse pas une certaine limite fixée.

## 2.4. Les objectifs

Les objectifs dits aussi les critères d'évaluation sont les indicateurs de performance sur lesquels se base le choix d'un ordonnancement satisfaisant. En ordonnancement, les critères à optimiser

consistent à minimiser ou maximiser une fonction objectif. Cette fonction objectif est généralement liée aux temps, aux ressources ou bien aux coûts.

1. Les objectifs liés au temps : c'est la catégorie des objectifs les plus étudiés en optimisation,

parmi les plus classiques, nous pouvons citer :

- Le temps total d'exécution : connue sous le nom de makespan et défini par  $C_{max} = \max_{i \in E} C_i$  ( $E$  désigne l'ensemble d'opération à ordonnancer) qui représente la date de fin du job le plus tardif. La minimisation de ce critère est souvent rencontrée puisque ça conduit à une meilleure utilisation de ressources (productivité).
- La somme des dates d'achèvement des jobs : on lui réfère aussi comme total flow time ou total completion time définie par  $\sum_{i \in E} C_i$ .
- Le retard algébrique maximal : connue sous le nom de lateness et défini par  $L_{max} = \max_{i \in E} L_i$ , tel que  $L_i = C_i - d_i$  est le retard algébrique pour chaque opération. L'objectif de la minimisation du retard algébrique maximal consiste donc à minimiser la quantité  $L_{max}$ .

2. Les objectifs liés aux ressources : les objectifs de ce type correspondent par exemple à :

- Maximisation de la charge d'une ressource e.g., maximiser l'utilisation de la machine ayant un bon rendement ou la machine la moins gourmand en énergie, etc.
- Minimisation de nombre de ressources nécessaires pour réaliser un ensemble d'opérations.

3. Les objectifs liés au coût : ces objectifs consistent généralement à minimiser les coûts de lancement, de production, de stockage, ou de transport.



### Complément

Généralement, l'objectif souhaité dans un problème d'ordonnement quelconque définit le critère à optimiser. Lorsque cet objectif renvoie un seul critère, il s'agit bien du cas d'optimisation mono-objectif, le cas le plus fréquent dans la littérature des problèmes d'ordonnement. Cependant, lorsque l'objectif renvoie simultanément plusieurs critères à la fois, nous parlons dans ce cas d'optimisation multi-objectif. Cette dernière est devenue plus en plus importante face à l'évolution et à la concurrence des systèmes de production.

## 3. Typologie des problèmes d'ordonnements

Problèmes à une opération	20
Problèmes à plus d'une opération	20
Exercice	21

Une typologie des problèmes d'ordonnement dans un atelier peut s'opérer selon le nombre et la nature des machines ainsi que l'ordre d'enchaînement des opérations (gamme de fabrication).

Deux grandes familles de problèmes d'ordonnement se présentent. La première famille regroupe les problèmes pour lesquels chaque job nécessite une seule opération. La deuxième regroupe ceux dont les jobs nécessitent plusieurs opérations.

### 3.1. Problèmes à une opération

En se basant sur la configuration des machines, nous distinguons pour la première catégorie :

- Problèmes à une machine : les problèmes d'atelier à une machine (single machine problem) consistent à ordonnancer, sur une seule machine, des jobs constitué d'une seule opération.
- Problèmes à machines parallèles : les problèmes d'atelier à machines parallèles (parallel machine problem) sont une généralisation des problèmes à une machine. Ce type d'atelier se caractérise par le fait que chaque opération peut être réalisée par n'importe quelle machine, disposée en parallèle, mais n'en nécessite qu'une seule. Le problème d'ordonnancement consiste donc à déterminer l'affectation des opérations aux machines puis le séquençement de ces opérations sur chaque machine.

Dans le dernier cas, il est possible de distinguer trois classes de machines :

- Machines parallèles identiques (identical parallel machines) : la durée d'exécution des opérations est la même sur toutes les machines.
- Machines parallèles uniformes (uniform parallel machines) : la durée d'exécution des opérations varie uniformément en fonction de la performance de la machine choisie.
- Machines parallèles indépendantes ou non liées (unrelated parallel machines) : les durées opératoires dépendent complètement des machines utilisées.

### 3.2. Problèmes à plus d'une opération

Les problèmes de la deuxième catégorie sont dits problèmes d'atelier du fait de la nécessité du passage de chaque job sur deux ou plusieurs machines dédiées. Suivant le mode de passage des opérations sur les différentes machines, trois types d'ateliers sont distingués à savoir :

- Problèmes Flow-Shop : les ateliers de type Flow-Shop appelés également ateliers à cheminement unique, il s'agit d'un ensemble de  $m$  machines disposées en séries. Toutes les opérations de tous les jobs passent par les machines dans le même ordre (flot unidirectionnel).
- Problèmes Job-Shop : dans cette classe d'ateliers, appelés aussi ateliers à cheminements multiples, chaque opération passe sur les machines dans un ordre fixé, mais à la différence du Flow-Shop, cet ordre peut être différent pour chaque job (flot multidirectionnel).
- Problèmes Open-Shop : dans cette classe d'ateliers, appelés aussi ateliers à cheminement libre, les gammes opératoires des différents jobs ne sont pas fixées a priori contrairement au problème d'atelier Job-Shop. Les opérations d'un même job peuvent donc être exécutées dans un ordre quelconque. Le problème consiste d'une part à déterminer le cheminement de chaque job et d'autre part à ordonnancer les jobs en tenant compte des gammes trouvées.

#### Remarque

Un cas particulier important est celui d'un Flow-Shop de permutation, le Flow-Shop est dit de permutation s'il existe une contrainte selon laquelle la séquence des opérations des différents jobs est la même sur chaque machine.



Champ	Sous champs		Notation
$\alpha$	$\alpha$ 1	Type de machine	{1, P, Q, R, F, J, O, FH, JF, OG}
	$\alpha$ 2	Nombre de machines	{ $\emptyset$ , m}
$\beta$	$\beta$ 1	Mode d'exécution des jobs	$\emptyset$ , <i>pmtn</i>
	$\beta$ 2	Ressources supplémentaire	{ $\emptyset$ , res}
	$\beta$ 3	Relation de précédence	{ $\emptyset$ , prec, tree }
	$\beta$ 4	Dates de disponibilité	{ $\emptyset$ , $r_i$ }
	$\beta$ 5	Durées opératoires	{ $\emptyset$ , $p_i = p$ }
	$\beta$ 6	Dates d'échéance	{ $\emptyset$ , $d_i$ }
	$\beta$ 7	Propriété d'attente	{ $\emptyset$ , nwt}
$\gamma$			$C_{max}, \sum w_i C_i, L_{max}, T_{max}, \sum w_i T_i$

Tableau 1 : Classification de Graham

Notation	Description
1	Problème à une seule machine
P	Problème à machines parallèles identiques
Q	Problème à machines parallèles uniformes
R	Problème à machines parallèles indépendantes
F	Flow-Shop
J	Job-Shop
O	Open-Shop
F	Flow-Shop hybride
J	Job-Shop flexible
OG	Open-Shop généralisé

Tableau 2 : Interprétation des notations du champ  $\alpha_1$

Notation	Description
pmtn	La préemption des opérations est autorisée
prec	Existence des contraintes de précédence entre les opérations
res	L'opération nécessite l'emploi d'une ou plusieurs ressources supplémentaires
nwt	Les opérations de chaque job doivent se succéder sans attente
$p_i = p$	Les temps d'exécution des tâches sont identiques et égaux à $p$
$r_i$	Une date de début au plus tôt est associée à chaque job $i$
$d_i$	Une date d'échéance est associée à chaque job $i$

Tableau 3 : Interprétation des principales notations possibles de sous-champs du champ  $\beta$

Notation	Description
$C_{max}$	La durée totale de l'ordonnancement
$L_{max}$	Le plus grand retard algébrique
$T_{max}$	Le plus grand retard vrai
$\sum [w_i]C_i$	La somme [pondéré] des dates de fin des tâches
$\sum [w_i]T_i$	La somme [pondéré] des retards
$\sum [w_i]U_i$	Le nombre [pondéré] des tâches en retard

Tableau 4 : Interprétation des principales notations du champ  $\gamma$ 

### Exemple

$Pm|pmtn|L_{max}$  désigne le problème de la minimisation du retard maximum  $L_{max}$  dans un environnement à  $m$  machines parallèles identiques où la préemption est autorisée.

## 5. Exercice

À quel type de problème correspond cette notation  $1|prec, r_i| \sum w_i C_i$  ?

## 6. La théorie de la complexité

Complexité algorithmique  
Complexité problématique

25  
25

Les problèmes d'optimisation sont des problèmes dont la résolution consiste à trouver parmi un ensemble de solutions celle qui répond le mieux à certains critères décrits sous forme d'une fonction objectif. Lorsque le domaine de solutions est discret, on parle alors de problèmes d'optimisation combinatoire. Les problèmes d'ordonnancement font partie des problèmes d'optimisation combinatoire.

Selon Cook, la théorie de la complexité a pour but d'analyser les coûts de résolution, notamment en terme de temps de calcul, des problèmes d'optimisation combinatoire.

La définition de la complexité d'un problème découle de la définition de la complexité des algorithmes. Informellement, la complexité d'un problème peut être définie comme la complexité du meilleur algorithme permettant sa résolution.

## 6.1. Complexité algorithmique

La complexité algorithmique est un concept fondamental qui permet de mesurer les performances d'un algorithme. Ces performances sont évaluées sur la base du temps alloué pour l'exécution de l'algorithme ou encore par rapport à l'espace mémoire requis pour résoudre le problème. Généralement, le temps d'exécution est le facteur dominant pour déterminer l'efficacité d'un algorithme, pour cela, on se concentre principalement sur ce facteur.

La complexité temporelle d'un algorithme représente le nombre maximum d'opérations élémentaires effectuées pour résoudre un problème donné. Cette complexité se base essentiellement sur la mesure d'un ordre de grandeur qui est évalué en fonction de la taille du problème  $n$ . La notation  $O$  est utilisé pour représenter cet ordre de grandeur. Par exemple, pour un algorithme donné, si la solution est donnée en environ  $n$  opérations, on dit que l'algorithme a une complexité en  $O(n)$ .

### Définition

---

Un algorithme est dit polynomiale si pour tout  $n$ , l'algorithme s'exécute en moins de  $nk$  opérations élémentaires ( $k$  étant une constante).

### Définition

---

Un algorithme est dit non-polynomial si le nombre d'opérations n'est pas borné par un polynôme de  $n$ .

## 6.2. Complexité problématique

La complexité problématique dépend de la difficulté du problème à résoudre et du nombre d'opérations élémentaires qu'un algorithme peut effectuer pour trouver l'optimum en fonction de la taille du problème.

Selon Garey et Johnson, la théorie de complexité d' problème d'optimisation se limite à la seule étude de problème de décision. Celui-ci, aussi appelé problème de reconnaissance, est un problème dont la résolution se limite à la réponse par « oui » ou par « non » à la question de savoir s'il existe une solution à un problème donné.

Pour chaque problème d'optimisation, on peut associer un problème de décision défini par la question suivante : « Existe-t-il une solution telle que son évaluation est majorée (Resp. minorée) par une constante fixée ? ». Ceci est pour un problème d'optimisation, dont la fonction objectif est la minimisation (resp. maximisation). Par exemple, dans un problème de type Flow-Shop  $Fm||C_{max}$  ayant comme fonction objectif la minimisation du makespan, le problème de décision associé est : « Existe-t-il un ordonnancement pour un Flow-Shop où le makespan est inférieur à une certaine valeur  $z$  ».

Alors si la réponse a la question existe et est obtenue grâce à un algorithme polynomial alors il existera un algorithme polynomial résolvant le problème d'optimisation associé. Cette propriété nous permet de classer les problèmes d'optimisation grâce à leurs problèmes de décision. Ces problèmes peuvent être classés en deux classes principales : la classe P et la classe NP.

- La classe P (Polynomiale) : les problèmes appartenant à la classe P sont ceux dont le problème de décision correspondant peut être résolu à l'optimum, à l'aide d'un algorithme en temps

polynomial. C'est en quelque sorte la classe des problèmes dits « facile ».

- La classe NP (Non-deterministic Polynomial) : cette classe a un nom trompeur, NP ne correspond pas à Non Polynomial, mais à Polynomial Non-déterministe ou en anglais « Nondeterministic Polynomial ». Cette classe est une extension de la classe P, elle représente la classe des problèmes de décision pour lesquels un algorithme non-déterministe peut vérifier en temps polynomial la validité d'une solution du problème traité.