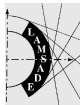


# Introduction aux problèmes d'ordonnancement

Mohamed Ali ALOULOU

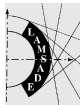
LAMSADE  
Université Paris Dauphine  
E-mail : [aloulou@lamsade.dauphine.fr](mailto:aloulou@lamsade.dauphine.fr)

3 novembre 2005



## Plan du cours

- 1** Définition et formulation du problème d'ordonnancement
- 2** Ordonnancement de projet : rappels et extensions
- 3** Ordonnancement d'ateliers : contexte et classification
- 4** Ordonnancement d'ateliers : méthodes de résolution



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

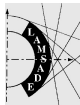
Contraintes en  
ordonnancement

Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Partie 1 : Définition et formulation du problème d'ordonnancement

- 1 C'est quoi l'ordonnancement ?
- 2 Quelques domaines concernés par la fonction ordonnancement
- 3 La fonction ordonnancement dans la gestion de la production
- 4 Contraintes rencontrées en ordonnancement
  - Contraintes de potentiel
  - Contraintes disjonctives
  - Contraintes cumulatives
- 5 Formulation mathématique



Plan

C'est quoi  
l'ordonnancement ?

Domaines

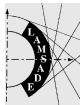
Ordonnancement en  
GdPContraintes en  
ordonnancementContraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulativesFormulation  
mathématique

# Ordonnancer ?

## Définition

Le problème d'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînements, ...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises.

- Un ensemble de **tâches**
  - Un environnement de **ressources** pour effectuer les tâches
  - Des **contraintes** sur les tâches et les ressources
  - Un **critère d'optimisation**
- ⇒ Déterminer les dates d'exécution des tâches



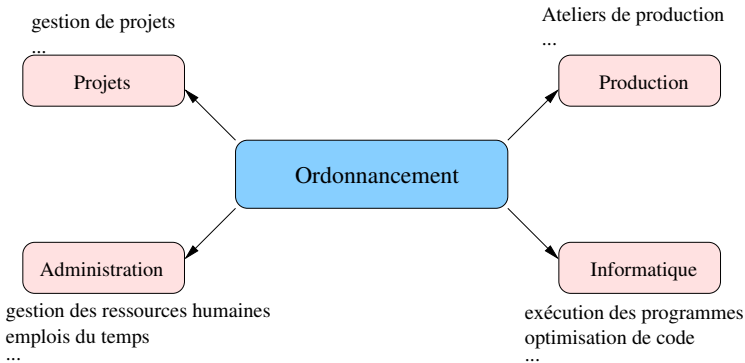
Plan

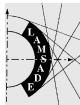
C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdPContraintes en  
ordonnancementContraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulativesFormulation  
mathématique

## Domaines concernés





## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement  
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

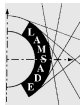
Formulation  
mathématique

## La gestion de la production [Giard 2003]

*La gestion de production a pour objet la recherche d'une organisation efficace de la production des biens et des services*

3 catégories pour classer les décisions en gestion de la production :

- les décisions **stratégiques** : politique long terme de l'entreprise
- les décisions **tactiques** : décisions à moyen terme
  - planification de la production
  - plan de transport
- les décisions **opérationnelles** : court terme
  - gestion des stocks
  - ordonnancement
  - pilotage informatique en temps réel



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

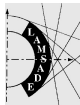
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Contraintes rencontrées en ordonnancement

### Différentes contraintes

- technologiques : une tâche ne peut débuter que lorsque d'autres sont achevées
- commerciales : certaines dates doivent être achevées pour une date fixée
- matérielles : une machine ne peut traiter qu'une machine à la fois
- de main d'oeuvre : effectif limité
- financières : budget limité



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

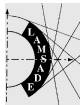
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Contraintes rencontrées en ordonnancement

### Différentes contraintes

- technologiques : une tâche ne peut débuter que lorsque d'autres sont achevées
- commerciales : certaines dates doivent être achevées pour une date fixée
- matérielles : une machine ne peut traiter qu'une machine à la fois
- de main d'oeuvre : effectif limité
- financières : budget limité



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

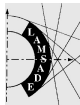
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Contraintes rencontrées en ordonnancement

### Différentes contraintes

- technologiques : une tâche ne peut débuter que lorsque d'autres sont achevées
- commerciales : certaines dates doivent être achevées pour une date fixée
- matérielles : une machine ne peut traiter qu'une machine à la fois
- de main d'oeuvre : effectif limité
- financières : budget limité



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

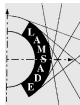
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Contraintes rencontrées en ordonnancement

### Différentes contraintes

- technologiques : une tâche ne peut débuter que lorsque d'autres sont achevées
- commerciales : certaines dates doivent être achevées pour une date fixée
- matérielles : une machine ne peut traiter qu'une machine à la fois
- de main d'oeuvre : effectif limité
- financières : budget limité



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

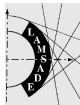
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Contraintes rencontrées en ordonnancement

### Différentes contraintes

- technologiques : une tâche ne peut débuter que lorsque d'autres sont achevées
- commerciales : certaines dates doivent être achevées pour une date fixée
- matérielles : une machine ne peut traiter qu'une machine à la fois
- de main d'oeuvre : effectif limité
- financières : budget limité



Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

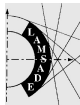
Contraintes en  
ordonnancement

Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Formalisation des contraintes

- 1 Contraintes potentielles (ou de potentiels)
- 2 Contraintes disjonctives
- 3 Contraintes cumulatives



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

**Contraintes de potentiel**  
Contraintes disjonctives  
Contraintes cumulatives

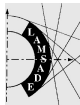
Formulation  
mathématique

## Contraintes potentielles

**Notation** :  $t_j$  : date de début de la tâche  $j$ ,  $p_j$  sa durée

**Forme générale** :  $t_j - t_i \geq a_{ij}$

- Localisation temporelle :  $j$  ne peut débuter avant une certaine date (livraison de matière première, conditions climatiques,...)
- Contrainte de délai :  $j$  doit être terminée avant une certaine date
- Contrainte de succession



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

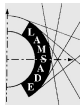
Formulation  
mathématique

## Contraintes potentielles

**Notation** :  $t_j$  : date de début de la tâche  $j$ ,  $p_j$  sa durée

**Forme générale** :  $t_j - t_i \geq a_{ij}$

- Localisation temporelle :  $j$  ne peut débuter avant une certaine date (livraison de matière première, conditions climatiques,...)
- Contrainte de délai :  $j$  doit être terminée avant une certaine date
- Contrainte de succession
  - succession simple
  - succession avec attente
  - succession avec chevauchement
  - succession immédiate



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement  
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

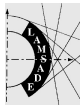
Formulation  
mathématique

## Contraintes potentielles

**Notation** :  $t_j$  : date de début de la tâche  $j$ ,  $p_j$  sa durée

**Forme générale** :  $t_j - t_i \geq a_{ij}$

- Localisation temporelle :  $j$  ne peut débuter avant une certaine date (livraison de matière première, conditions climatiques,...)
- Contrainte de délai :  $j$  doit être terminée avant une certaine date
- Contrainte de succession
  - succession simple
  - succession avec attente
  - succession avec chevauchement
  - succession immédiate



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

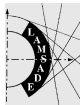
Formulation  
mathématique

## Contraintes potentielles

**Notation** :  $t_j$  : date de début de la tâche  $j$ ,  $p_j$  sa durée

**Forme générale** :  $t_j - t_i \geq a_{ij}$

- Localisation temporelle :  $j$  ne peut débuter avant une certaine date (livraison de matière première, conditions climatiques,...)
- Contrainte de délai :  $j$  doit être terminée avant une certaine date
- Contrainte de succession
  - succession simple
  - succession avec attente
  - succession avec chevauchement
  - succession immédiate



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

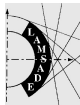
## Contraintes disjonctives

Deux tâches  $i$  et  $j$  sont en disjonction si elles ne peuvent être exécutées simultanément

⇒ Les intervalles d'exécution des tâches disjonctives sont disjoints :  $]t_i, t_i + p_i[ \cap ]t_j, t_j + p_j[ = \emptyset$

Disjonction d'inégalités de potentiels

$$t_j - t_i \geq p_i \quad \text{ou} \quad t_i - t_j \geq p_j$$



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement  
Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Contraintes cumulatives

C'est une généralisation des contraintes disjonctives

**Exemple** : On a deux grues et 5 tâches nécessitant une grue sont candidates au même moment

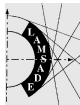
Soit

- $w_k(t)$  la quantité de moyen  $k$  disponible à  $t$ .
- $w_{ik}(t)$  la quantité de moyen  $k$  nécessaire pour exécuter  $i$  à  $t$ .

Si

$$\sum_{i \in S} w_{ik}(t) \leq w_k(t)$$

**alors** les tâches de  $S$  peuvent être exécutées simultanément à  $t$   
**sinon** les tâches de  $S$  sont en disjonction



## Plan

C'est quoi  
l'ordonnancement ?

Domaines

Ordonnancement en  
GdP

Contraintes en  
ordonnancement

Contraintes de potentiel  
Contraintes disjonctives  
Contraintes cumulatives

Formulation  
mathématique

## Formulation mathématique

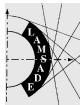
$n$  tâches à exécuter + 2 tâches fictives 0 et  $n + 1$  de durées nulles.

Déterminer  $(t_0, t_1, \dots, t_n, t_{n+1})$  de façon à  
Minimiser  $f(t_0, t_1, \dots, t_n, t_{n+1})$

s.c.

- 1 Contraintes de potentiel :  $t_j - t_i \geq a_{ij}$
- 2 Contraintes disjonctives :  $t_j - t_i \geq p_i$  ou  $t_i - t_j \geq p_j$
- 3 Contraintes cumulatives (idem)
- 4 Contraintes de non négativité :  $t_0, t_1, \dots, t_n, t_{n+1} \geq 0$

$f$  est fonction des dates de début (ou de fin des tâches), par exemple,  $f(t_0, t_1, \dots, t_n, t_{n+1}) = t_{n+1} - t_0$



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Partie 2 : Ordonnancement de projet – rappels et extensions

- 6** Problème central de l'ordonnancement : ressources illimitées
  - Définition
  - Modélisation avec un graphe potentiels-tâches
  - Recherche d'ordonnancement admissible
- 7** Cas général : ressources limitées
  - Problématique
  - Résolution exacte
  - Approche simple de résolution : algorithme de liste
- 8** Cas de ressources financières
  - Problématique
  - L'offre et la demande
  - Algorithme de décalage



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

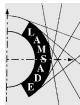
Problématique  
L'offre et la demande  
Algorithme de décalage

## Contexte

Un projet consiste en un ensemble de  $n$  tâches liées par des contraintes de succession ou de précedence

## Objectif

- Calculer la durée minimale du projet, les ressources étant supposées illimitées  
 $\implies$  Minimiser  $(t_{n+1} - t_0)$  sous les contraintes de potentiels
- Déterminer les dates de début au plus tôt et au plus tard des tâches
- Déterminer les tâches critiques



## Plan

Problème central de l'ordonnement : ressources illimitées

## Définition

Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

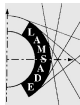
Problématique  
L'offre et la demande  
Algorithme de décalage

## Formulation mathématique

Déterminer  $(t_0, t_1, \dots, t_n, t_{n+1})$  de façon à  
Minimiser  $(t_{n+1} - t_0)$

s.c.

- 1** Contraintes de potentiel :  $t_j - t_i \geq a_{ij}$
- 2** Contraintes de non négativité :  $t_0, t_1, \dots, t_n, t_{n+1} \geq 0$



## Plan

Problème central de l'ordonnement : ressources illimitées

## Définition

Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Formulation mathématique

Déterminer  $(t_0, t_1, \dots, t_n, t_{n+1})$  de façon à  
Minimiser  $(t_{n+1} - t_0)$

s.c.

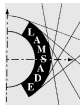
**1** Contraintes de potentiel :  $t_j - t_i \geq a_{ij}$

**2** Contraintes de non négativité :  $t_0, t_1, \dots, t_n, t_{n+1} \geq 0$

C'est un PL

MAIS, IL Y A PLUS SIMPLE !

⇒ **Méthode potentiels-tâches** et méthode PERT



## Plan

Problème central de  
l'ordonnement :  
ressources illimitées

Définition

**Modélisation**

Résolution

Cas général :  
ressources limitées

Problématique

Résolution exacte

Algorithme de liste

Cas de ressources  
financières

Problématique

L'offre et la demande

Algorithme de décalage

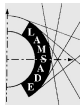
## Modélisation avec un graphe potentiels-tâches

On utilise un graphe orienté  $G = (X, U)$

- $X$  : ensemble des sommets : tâches
- $U$  : ensemble des arcs : contraintes de potentiels

$$U = \{(i, j) \in X \times X, \exists \text{ contrainte } t_j - t_i \geq a_{ij}\}$$

La valuation d'un arc  $(i, j)$  est  $v_{ij} = a_{ij}$



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

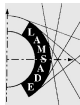
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Exemple illustratif

Rubrique	Tâche	Description	Durée	Tâches préc.
	Début	Lancement du projet	0	-
Sols	A	Dépose ancien carrelage	6	J
	B	Pose carrelage	4	A
	C	Joints carrelage	2	B
Murs	D	Décollage ancien papier	8	J
	E	Pose faïence	6	D
	F	Pose nouveau papier	4	E
Plomberie	G	Dépose ancien évier	1	Début
	H	déplacement arriv. et évac.	6	G
	I	Pose et raccordement Evier	2	M
Mobilier	J	Dépose ancien éléments	4	G
	K	Assemb. caissons et tiroirs	8	Début
	L	Pose éléments bas	6	B,J,H,K
	M	Pose plan de travail	4	L
	N	Etanchéité plan de travail	1	E,M
	O	Pose éléments hauts	1	E,J
	Fin	Fin du projet	0	C,F,I,N,O



## Plan

Problème central de l'ordonnement : ressources illimitées

Définition

**Modélisation**

Résolution

Cas général : ressources limitées

Problématique

Résolution exacte

Algorithme de liste

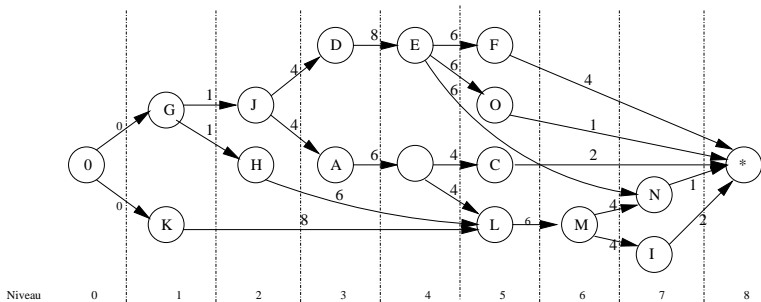
Cas de ressources financières

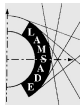
Problématique

L'offre et la demande

Algorithme de décalage

## Graphe correspondant





## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

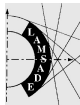
Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

# Recherche d'un ordonnancement réalisable

## Théorème

*Il existe un ordonnancement réalisable ssi  $\nexists$  de circuit de valeur strictement positive dans  $G$*



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Recherche d'un ordonnancement réalisable

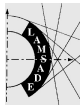
### Théorème

*Il existe un ordonnancement réalisable ssi  $\nexists$  de circuit de valeur strictement positive dans  $G$*

Si la condition d'existence est vérifiée alors, il existe en général plusieurs ordonnancements réalisables de durée minimale.

On distinguera deux cas particuliers :

- 1** Ordonnancement au plus tôt
- 2** Ordonnancement au plus tard (avec une date limite d'achèvement du projet imposée)



## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

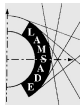
Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Dates de début au plus tôt (1)

### Définition

La date de début au plus tôt  $t_i$  d'une tâche  $i$  est égale à la **longueur du plus long chemin** de la tâche Début (ou 0) à  $i$ .



## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Dates de début au plus tôt (1)

### Définition

La date de début au plus tôt  $\underline{t}_i$  d'une tâche  $i$  est égale à la longueur du plus long chemin de la tâche Début (ou 0) à  $i$ .

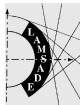
### Théorème

Dans le cas d'un graphe sans circuit, on a

$$\underline{t}_0 = 0$$

$$\underline{t}_i = \max_{j \in \Gamma^{-1}(i)} (\underline{t}_j + v_{j,i})$$

$v_{j,i}$  est la valuation de l'arc  $(j, i)$  (par exemple la durée de  $j$ )  
 $\Gamma^{-1}(i)$  est l'ensemble des prédécesseurs de  $i$ .



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

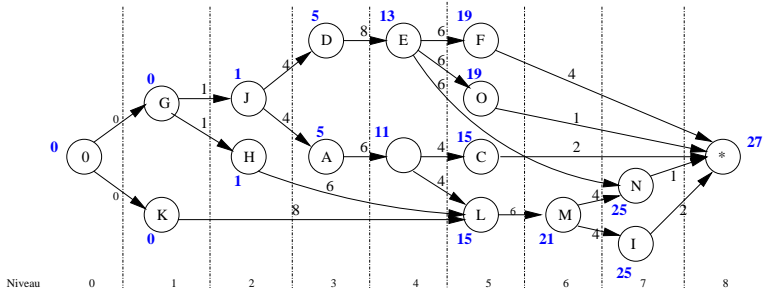
Cas général : ressources limitées

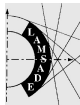
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Dates de début au plus tôt (2)





## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Dates de début au plus tard (1)

On souhaite terminer le projet au plus tard à une date

$$D \geq t_{n+1}$$

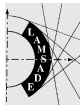
### Définition

La date de début au plus tard  $\bar{t}_i$  de  $i$  est la date maximum à laquelle on peut exécuter  $i$  sans retarder le chantier (date  $D$ ).

### Théorème

$$\bar{t}_{n+1} = D$$

$$\bar{t}_i = \bar{t}_{n+1} - \text{valeur d'un plus long chemin entre } i \text{ et } n + 1$$



## Plan

Problème central de  
l'ordonnement :  
ressources illimitées

Définition  
Modélisation  
Résolution

Cas général :  
ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources  
financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Dates de début au plus tard (2)

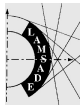
### Théorème

*Dans le cas d'un graphe sans circuit, on a*

$$\bar{t}_* = D$$

$$\bar{t}_i = \min_{j \in \Gamma(i)} \{\bar{t}_j - v_{i,j}\}$$

$v_{i,j}$  est la valuation de l'arc  $(i, j)$  (par exemple la durée de  $i$ )  
 $\Gamma(i)$  est l'ensemble des successeurs de  $i$ .



## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

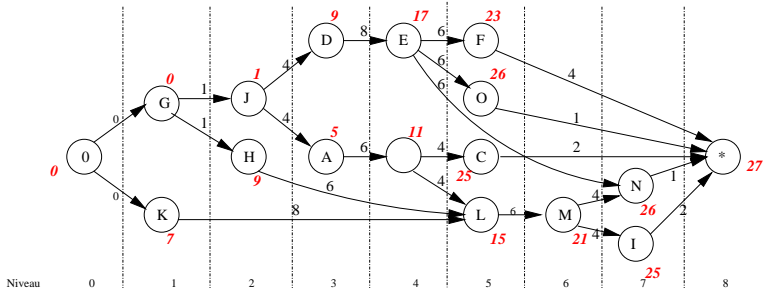
Cas général : ressources limitées

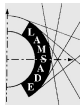
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Dates de début au plus tard (3)





## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Marges totales et chemin critique (1)

### Définition

La **marge totale** d'une tâche  $i$  est le retard total qu'on peut se permettre sur  $i$  sans remettre en cause la date de fin du projet.

$$MT_i = \bar{t}_i - \underline{t}_i$$

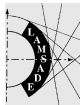
### Définition

Les **tâches critiques** ont une marge nulle (par extension si

$$MT_i = MT_*)$$

⇒ Tout retard sur leur exécution entraîne un retard global sur le projet.

Un **chemin est critique** s'il relie Début à Fin et s'il ne contient que des tâches critiques.



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

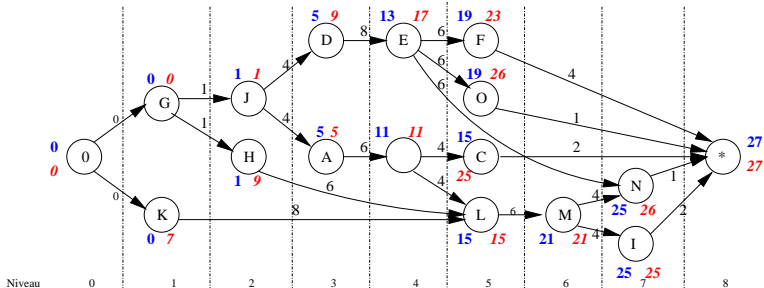
Cas général : ressources limitées

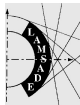
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Marges totales et chemin critique (2)





## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

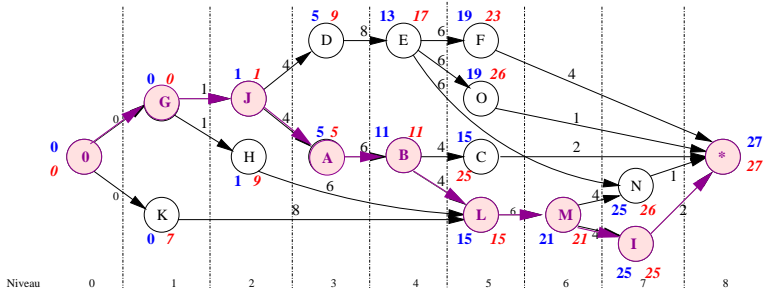
Cas général : ressources limitées

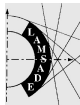
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Marges totales et chemin critique (3)





## Plan

Problème central de  
l'ordonnancement :  
ressources illimitées

Définition  
Modélisation  
Résolution

Cas général :  
ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

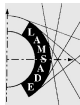
Cas de ressources  
financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Marges libres

### Définition

La **marge libre** d'une tâche  $i$  est le retard total qu'on peut se permettre sur  $i$  sans retarder l'exécution d'une autre tâche (par rapport aux dates de début au plus tôt).



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Marges libres

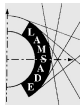
### Définition

La **marge libre** d'une tâche  $i$  est le retard total qu'on peut se permettre sur  $i$  sans retarder l'exécution d'une autre tâche (par rapport aux dates de début au plus tôt).

### Théorème

$$ML_i = \min_{j \in \Gamma(i)} \{ \underline{t}_j - \underline{t}_i - v_{i,j} \}$$

$v_{i,j}$  est la valuation de l'arc  $(i, j)$  (par exemple la durée de  $i$ )  
 $\Gamma(i)$  est l'ensemble des successeurs de  $i$ .



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition

Modélisation

Résolution

Cas général : ressources limitées

Problématique

Résolution exacte

Algorithme de liste

Cas de ressources financières

Problématique

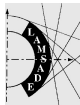
L'offre et la demande

Algorithme de décalage

# Marges libres (1)

## Définition

La **marge libre** d'une tâche  $i$  est le retard total qu'on peut se permettre sur  $i$  sans retarder l'exécution d'une autre tâche (par rapport aux dates de début au plus tôt).



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition

Modélisation

Résolution

Cas général : ressources limitées

Problématique

Résolution exacte

Algorithme de liste

Cas de ressources financières

Problématique

L'offre et la demande

Algorithme de décalage

## Marges libres (1)

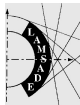
### Définition

La **marge libre** d'une tâche  $i$  est le retard total qu'on peut se permettre sur  $i$  sans retarder l'exécution d'une autre tâche (par rapport aux dates de début au plus tôt).

### Théorème

$$ML_i = \min_{j \in \Gamma(i)} \{ \underline{t}_j - \underline{t}_i - v_{i,j} \}$$

$v_{i,j}$  est la valuation de l'arc  $(i, j)$  (par exemple la durée de  $i$ )  
 $\Gamma(i)$  est l'ensemble des successeurs de  $i$ .



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

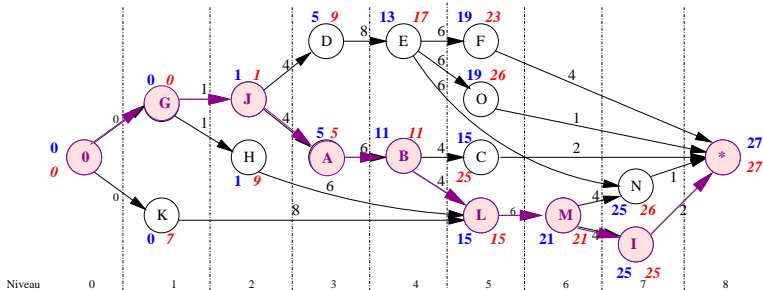
Cas général : ressources limitées

**Problématique**  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

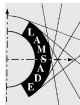
Problématique  
L'offre et la demande  
Algorithme de décalage

## Problématique



La tâches *D* et *A* doivent être effectuées par la même personne.

⇒ On doit choisir d'effectuer *D* avant *A* ou bien *A* avant *D*



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

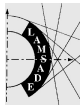
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Schéma général

- 1 Enumérer les différentes possibilités :
  - Si  $A$  est avant  $D$  alors on rajoute un arc de  $A$  vers  $D$  valué par la durée de  $A$ ,
  - Si  $D$  est avant  $A$  alors on rajoute un arc de  $D$  vers  $A$  valué par la durée de  $D$
- 2 Résoudre les problèmes correspondants
- 3 Choisir la meilleure décision



## Plan

Problème central de  
l'ordonnancement :  
ressources illimitées

Définition  
Modélisation  
Résolution

Cas général :  
ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

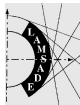
Cas de ressources  
financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Schéma général

- 1 Enumérer les différentes possibilités :
  - Si  $A$  est avant  $D$  alors on rajoute un arc de  $A$  vers  $D$  valué par la durée de  $A$ ,
  - Si  $D$  est avant  $A$  alors on rajoute un arc de  $D$  vers  $A$  valué par la durée de  $D$
- 2 Résoudre les problèmes correspondants
- 3 Choisir la meilleure décision

Lorsque le nombre de tâches partageant les mêmes ressources est trop important le problème devient très difficile à résoudre car le nombre de possibilités devient énorme !



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

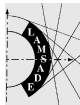
Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Approche simple de résolution

### Algorithme de liste

- Les conflits entre les tâches utilisant la même ressource sont résolus par des **règles de priorité statiques ou dynamiques**
- Exemples de règles de priorité
  - Plus petite date de début au plus tard d'abord
  - Plus petite date de fin au plus tard d'abord
  - Plus petite durée d'abord
- Les tâches sont ordonnancées en incrémentant le temps à partir de 0.



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

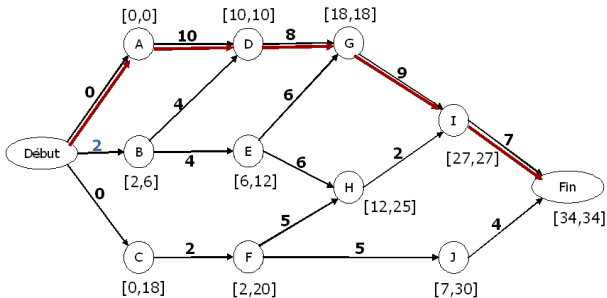
Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

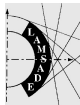
Problématique  
L'offre et la demande  
Algorithme de décalage

## Autre exemple



	A	B	C	D	E	F	G	H	I	J	Disponibilit
Ressource R1	3	3	1	1	1	2	3	2	1	2	5
Ressource R2	0	0	0	1	1	1	0	1	0	0	1

Appliquer l'algorithme de liste en donnant la priorité à la tâche qui possède la plus petite date de début au plus tard.



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

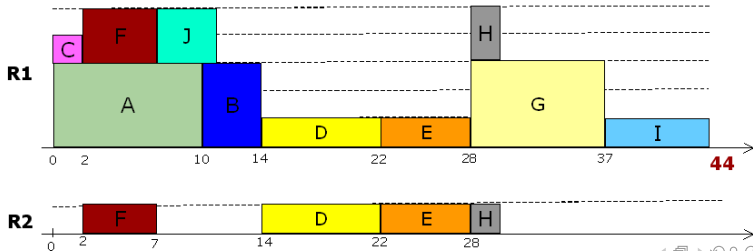
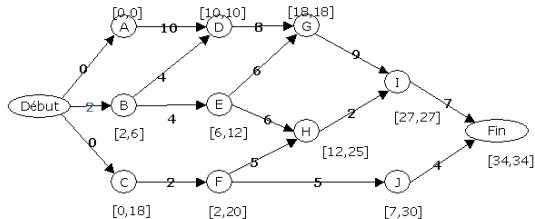
Cas général : ressources limitées

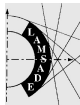
Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Diagramme de Gantt





Plan

Problème central de l'ordonnement : ressources illimitées

- Définition
- Modélisation
- Résolution

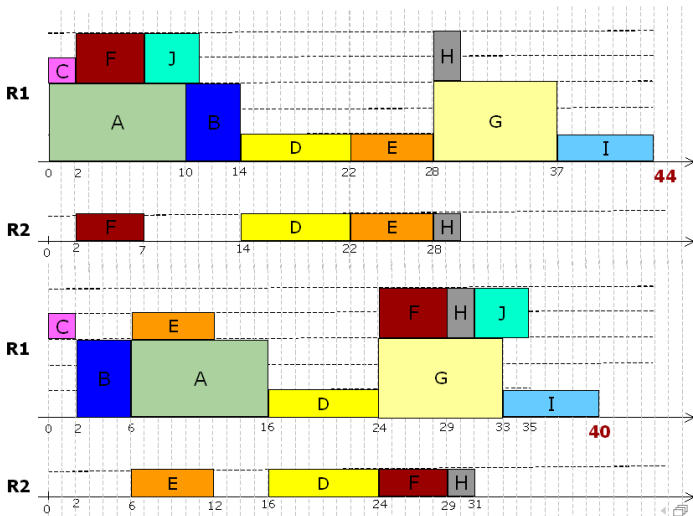
Cas général : ressources limitées

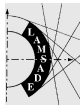
- Problématique
- Résolution exacte
- Algorithme de liste

Cas de ressources financières

- Problématique
- L'offre et la demande
- Algorithme de décalage

# Solution trouvée vs solution optimale





## Plan

Problème central de  
l'ordonnement :  
ressources illimitées

Définition  
Modélisation  
Résolution

Cas général :  
ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources  
financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Problématique

- Les tâches  $\{1, 2, \dots, n\}$  ont besoin d'une seule ressource :  
l'**argent**
  - La ressource est alimentée à des dates  $u_1, \dots, u_q$  et selon des quantités  $b_1, \dots, b_q$
  - Une tâche  $i$  requiert, à sa date de début d'exécution, une quantité  $a_i$  de la ressource
- Ordonnement de durée minimale en respectant les
  - contraintes de ressource
  - contraintes de précédence entre tâches



## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

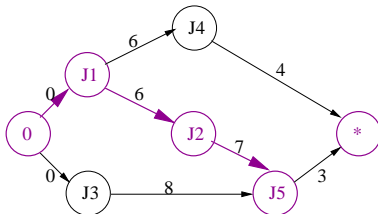
Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

Cas de ressources financières

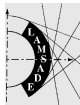
Problématique  
L'offre et la demande  
Algorithme de décalage

## Exemple



- **Demande** :  $\forall i \in \{J1, J2, J3, J4, J5\}, a_i = 5$
- **Offre**

$u_i$	$b_i$
0	15
10	5
16	10



## Plan

Problème central de l'ordonnancement : ressources illimitées

Définition  
Modélisation  
Résolution

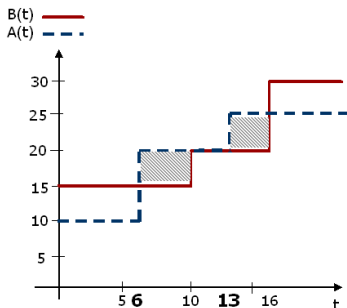
Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

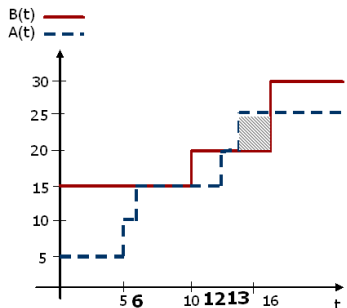
Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## L'offre et la demande

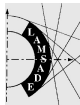


L'offre et la demande pour l'ordonnancement au plus tôt



L'offre et la demande pour l'ordonnancement au plus tard

Aucun des ordonnancements n'est admissible



## Plan

Problème central de l'ordonnement : ressources illimitées

Définition  
Modélisation  
Résolution

Cas général : ressources limitées

Problématique  
Résolution exacte  
Algorithme de liste

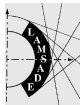
Cas de ressources financières

Problématique  
L'offre et la demande  
Algorithme de décalage

## Algorithme de décalage

- On peut montrer que si un ordonnancement est admissible, tout ordonnancement au plus tard de même durée l'est aussi
- **Algorithme de décalage**
  - Calculer un ordonnancement au plus tard
  - S'il n'est pas admissible, calculer la quantité minimale  $\delta^*$  dont il faut décaler l'ordonnancement au plus tard pour que l'ordonnancement résultant soit admissible

⇒ L'algorithme de décalage permet d'obtenir un ordonnancement optimal



Plan

Contexte

Classification

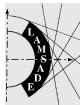
Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

Complexité ...  
... des algorithmes  
... des problèmes

## Partie 3 : Ordonnancement d'ateliers – contexte et classification

- 9** Contexte
- 10** Classification des problèmes d'ordonnancement
  - Schémas de classification
  - Les environnements machines
  - Les tâches et leurs caractéristiques
  - Les critères d'optimisation
- 11** Les différentes classes d'ordonnancement
- 12** Quelques notions de la théorie de la complexité
  - Complexité des algorithmes
  - Complexité des problèmes



Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

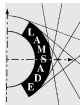
Les différentes classes d'ordonnancement

Complexité ...

... des algorithmes  
... des problèmes

## Contexte

- On a un ensemble  $\mathcal{J}$  de tâches ou de travaux (ou jobs) à exécuter
- Les ressources sont des machines et ne peuvent exécuter qu'une tâche à la fois
- Les critères font intervenir les dates de fin d'exécution, les dates de livraison, les stocks d'en cours ... et les ordres de fabrication peuvent avoir des poids (importance) différents



Plan

Contexte

Classification

Schémas de classification

Les environnements machines

Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes d'ordonnancement

Complexité ...

... des algorithmes

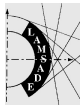
... des problèmes

## Schémas de classification

Nous suivons les schémas de classification proposés par (Graham et al, 1979).

Classification à trois champs  $\alpha|\beta|\gamma$

- $\alpha$  : environnement machine
- $\beta$  : les caractéristiques des tâches
- $\gamma$  : le (ou les) critère(s) à optimiser



## Plan

## Contexte

## Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

## Complexité ...

... des algorithmes  
... des problèmes

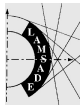
## D'une façon générale ...

On doit exécuter  $n$  tâches ou  $n$  travaux (*jobs*).

Le champ  $\alpha$  est décomposé en deux sous-champs  $\alpha_1$  et  $\alpha_2$ .

Selon les valeurs prises par  $\alpha_1$ , on distingue :

- Les problèmes à une machine
- Les problèmes à machines parallèles
- Les problèmes d'ateliers
  - Ateliers à cheminement unique (*flowshop*)
  - Ateliers à cheminements multiples (*jobshop*)
  - Ateliers à cheminements libres (*openshop*)
  - ...
- L'ordonnancement de projet sous contraintes de ressources



Plan

Contexte

Classification

Schémas de  
classification**Les environnements  
machines**Caractéristiques de  
tâchesLes critères  
d'optimisationLes différentes  
classes  
d'ordonnancement

Complexité ...

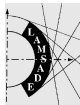
... des algorithmes

... des problèmes

## Problèmes à une machine

Toute tâche  $J_j, j \in \{1, \dots, n\}$  de durée  $p_j$  (*processing time*) s'exécute sur une machine qui ne peut traiter plus qu'une tâche à la fois.

Le champ  $\alpha_1$  est absent et  $\alpha_2 = 1$ .



Plan

Contexte

Classification

Schémas de classification

Les environnements machines

Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes

d'ordonnement

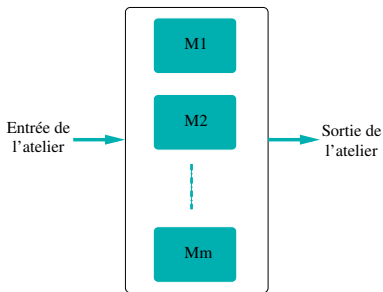
Complexité ...

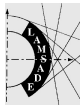
... des algorithmes

... des problèmes

## Problèmes à machines parallèles (1)

Toute tâche  $J_j, j \in \{1, \dots, n\}$  peut être exécutée indifféremment sur une des  $m$  machines mises en parallèle.





## Plan

## Contexte

## Classification

Schémas de classification

Les environnements machines

Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes

d'ordonnancement

## Complexité ...

... des algorithmes

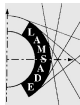
... des problèmes

## Problèmes à machines parallèles (2)

$p_{i,j}$  est la durée d'exécution de  $J_j$  sur la machine

$M_i, i = 1, \dots, m.$

- si  $\alpha_1 = P$  alors **machines identiques**  $\Rightarrow \forall i, p_{i,j} = p_j$
- si  $\alpha_1 = Q$  alors **machines uniformes**  $\Rightarrow \forall i, p_{i,j} = p_j/s_i$  où  $s_i$  est la vitesse de traitement  $M_i$
- si  $\alpha_1 = R$  alors **machines indépendantes**  $\Rightarrow \forall i, p_{i,j} = p_j/s_{i,j}$  où  $s_{i,j}$  est la vitesse de traitement de la tâche  $J_j$  par la machine  $M_i$
- si  $\alpha_2$  est un entier positif, le nombre de machines est supposé constant. Si  $\alpha_2$  est absent alors ce nombre est supposé arbitraire.



## Plan

## Contexte

## Classification

Schémas de classification

Les environnements machines

Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes

d'ordonnement

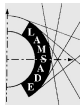
Complexité ...

... des algorithmes

... des problèmes

## Problèmes d'ateliers

- $m$  machines différentes  $M_i, i \in \{1, \dots, m\}$
- $n$  travaux (jobs)  $J_j, j \in \{1, \dots, n\}$ .
  - Chaque job  $J_j$  est décrit par  $n_j$  tâches ou **opérations**  $O_{i,j}, i \in \{1, \dots, n_j\}$
  - La durée d'une opération  $O_{i,j}$  est  $p_{i,j}$
  - La machine qui exécute l'opération  $O_{i,j}$  du job est notée  $M(O_{i,j})$  ou  $M_{i,j}$ .
  - **Les opérations d'un même job ne peuvent pas être exécutées simultanément**
- $\alpha_2$  (voir machines parallèles)



## Plan

## Contexte

## Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

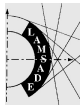
## Complexité ...

... des algorithmes  
... des problèmes

## Ateliers à cheminement unique : Flowshop



- Chaque Job est constitué de  $m$  opérations et l'ordre de passage sur les différentes machines est le même pour tous les jobs  $J_j : O_{1,j} \rightarrow O_{2,j} \rightarrow, \dots, \rightarrow O_{m,j}$  et  $M_{i,j} = M_i$
- $\alpha_1 = F$



## Plan

## Contexte

## Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnement

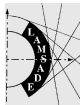
## Complexité ...

... des algorithmes  
... des problèmes

## Ateliers à cheminements quelconques : Jobshop

- Le nombre d'opérations n'est pas forcément le même pour tous les jobs
- Chaque job a son propre ordre de passage sur les machines
- $\alpha_1 = J$
- Exemple

$J_j$	$J_1$			$J_2$			$J_3$	
$O_{i,j}$	$O_{1,1}$	$O_{2,1}$	$O_{3,1}$	$O_{1,2}$	$O_{2,2}$	$O_{3,2}$	$O_{1,3}$	$O_{2,3}$
$M_{i,j}$	$M_1$	$M_2$	$M_3$	$M_2$	$M_1$	$M_3$	$M_3$	$M_2$
$p_{i,j}$	3	2	5	4	2	2	2	3



Plan

Contexte

Classification

Schémas de classification

**Les environnements machines**

Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes d'ordonnancement

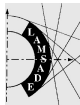
Complexité ...

... des algorithmes

... des problèmes

## Ateliers à cheminements libres : Openshop

- Le nombre d'opérations n'est pas forcément le même pour tous les jobs
- L'ordre de passage sur les machines est totalement libre
- $\alpha_1 = O$



Plan

Contexte

Classification

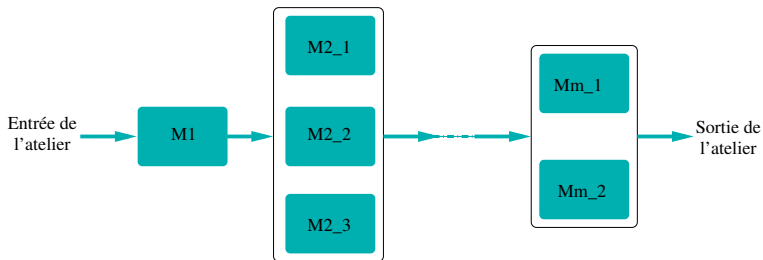
Schémas de  
classification**Les environnements  
machines**Caractéristiques de  
tâchesLes critères  
d'optimisationLes différentes  
classes  
d'ordonnancement

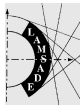
Complexité ...

... des algorithmes

... des problèmes

## Autres ateliers : Flowshop hybride





## Plan

## Contexte

## Classification

Schémas de classification  
Les environnements machines

## Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes d'ordonnancement

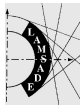
## Complexité ...

... des algorithmes  
... des problèmes

## Les tâches et leurs caractéristiques

Chaque tâche  $J_j$  ou chaque job  $J_j$  peut être caractérisé par

- une date de début au plus tôt  $r_j$  (**release date**)
- une durée  $p_j$ , pour la tâche, ou  $p_{i,j}$ , pour l'opération  $i$  du job (**processing time**)
- une date de fin souhaitée  $d_j$  (**due date**)
- une date de fin obligatoire  $\tilde{d}_j$  (**deadline**)
- un poids relatif  $w_j$  (**weight**)  $\Rightarrow$  importance ou poids
- Il peut y avoir des contraintes de précédence entre les tâches. Ces contraintes sont représentées par un graphe  $G = (\mathcal{J}, A)$ .



## Plan

## Contexte

## Classification

Schémas de classification  
Les environnements machines

## Caractéristiques de tâches

Les critères d'optimisation

Les différentes classes d'ordonnement

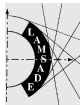
## Complexité ...

... des algorithmes  
... des problèmes

## Les tâches et leurs caractéristiques

$\beta = \beta_1\beta_2\beta_3\beta_4 \dots$  dans  $\alpha|\beta|\gamma$

- $\beta_1 = pmtn$  si la **préemption** des tâches est autorisée, sinon  $\beta_1$  est absent
- S'il y a des **contraintes de précédence** entre les tâches  $\beta_2 \in \{prec, chain, in - tree, out - tree\}$ , sinon  $\beta_2$  est vide
- $\beta_3 = r_j$  si les **dates de début au plus tôt**  $r_j$  (ou dates de disponibilité) des tâches ne sont pas forcément identiques, sinon  $(\forall j, r_j = 0)$   $\beta_3$  est absent
- $\beta_4 = \tilde{d}_j$  si la tâche ou le job possède une **date de fin obligatoire**



## Plan

## Contexte

## Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnement

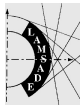
## Complexité ...

... des algorithmes  
... des problèmes

## Quelques mesures

Les critères d'optimisation s'expriment en fonction des dates de fin des tâches (ou jobs)  $C_j$  (**completion times**). Les critères sont généralement exprimés en fonction des mesures suivantes

- le retard algébrique  $L_j = C_j - d_j$  (**lateness**)
- le retard absolu  $T_j = \max(0, C_j - d_j)$  (**tardiness**)
- l'avance  $E_j = \max(0, d_j - C_j)$  (**earliness**)
- le pénalité unitaire de retard  $U_j = 0$  si  $C_j \leq d_j$ ,  $U_j = 1$  sinon
- le durée de séjour dans l'atelier  $F_j = C_j - r_j$
- une fonction générique  $f_j(t)$  donnant le coût induit si on termine  $J_j$  à  $t$



Plan

Contexte

Classification

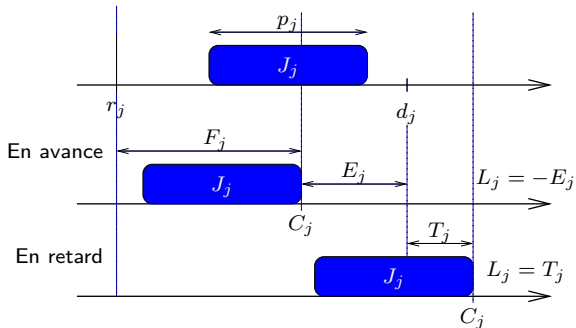
- Schémas de classification
- Les environnements machines
- Caractéristiques de tâches
- Les critères d'optimisation

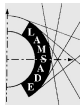
- Les différentes classes d'ordonnancement

Complexité ...

- ... des algorithmes
- ... des problèmes

## Illustration





Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

Complexité ...

... des algorithmes  
... des problèmes

## Les critères d'optimisation

On cherche alors à minimiser un ou plusieurs critères

$F(C_1, \dots, C_n)$

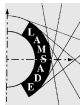
- une fonction générique de **coût maximum**

$$f_{\max} = \max_j \{f_j(C_j)\}$$

- la durée totale de l'ordonnancement  $C_{\max} = \max_j \{C_j\}$
- le retard algébrique maximum  $L_{\max} = \max_j L_j$
- le retard maximum  $T_{\max} = \max_j T_j$

- une fonction générique de **coût total**  $f_{\Sigma} = \sum_j \{f_j(C_j)\}$

- la somme (pondérée) des dates de fin  $\sum_j (w_j)C_j$
- la somme (pondérée) des retards  $\sum_j (w_j)T_j$
- le nombre (pondéré) des tâches (ou jobs) en retard  $\sum_j (w_j)U_j$
- la durée moyenne de séjour  $\sum_j F_j$
- la somme (pondérée) des avances et des retards  $\sum_j \alpha_j E_j + \beta_j T_j$



Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

Complexité ...

... des algorithmes  
... des problèmes

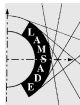
## Critères réguliers

### Définition

Un **critère**  $F(C_1, \dots, C_n)$  est dit **régulier** si et seulement si  $F$  est une fonction croissante des dates de fin des tâches (ou jobs)

### Corollaire

- Les critères  $C_{\max}$ ,  $L_{\max}$ ,  $T_{\max}$ ,  $\sum_j (w_j)C_j$ ,  $\sum_j (w_j)T_j$ ,  $\sum_j (w_j)U_j$ ,  $\sum_j (w_j)F_j$  sont réguliers
- Le critère  $\sum_j \alpha_j E_j + \beta_j T_j$  n'est pas régulier



Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

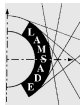
Complexité ...

... des algorithmes  
... des problèmes

## Les différentes classes d'ordonnancement

Quatre catégories ou classes (par ordre d'inclusion)

- les ordonnancements quelconques
- les ordonnancements **semi-actifs**
- les ordonnancements **actifs**
- les ordonnancements **sans délai**



Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnement

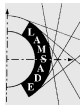
Complexité ...

... des algorithmes  
... des problèmes

# Les différentes classes d'ordonnement

## Définitions

- Un ordonnancement est dit **semi-actif** si aucune tâche ne peut être exécutée plus tôt sans changer l'ordre d'exécution sur les ressources ou violer de contrainte (de précédence, date de début au plus tôt, ...)
- Un ordonnancement est dit **actif** si aucune tâche ne peut être exécutée plus tôt sans retarder une autre tâche ou violer de contrainte
- Un ordonnancement **sans délai** est un ordonnancement pour lequel aucune machine n'est laissée inactive alors qu'elle pourrait commencer une tâche disponible



## Plan

## Contexte

## Classification

- Schémas de classification
- Les environnements machines
- Caractéristiques de tâches
- Les critères d'optimisation

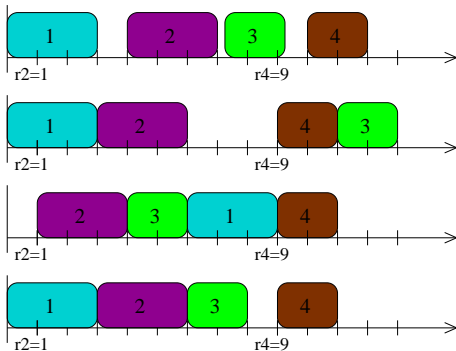
## Les différentes classes d'ordonnement

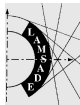
## Complexité ...

- ... des algorithmes
- ... des problèmes

## Les différentes classes d'ordonnement

**Exemple** : 4 tâches, à exécuter sur une machine, telles que  $r_1 = r_3 = 0$ ,  $r_2 = 1$ ,  $r_4 = 9$ ;  $p_1 = p_2 = 3$ ,  $p_3 = 2$ ,  $p_4 = 2$





Plan

Contexte

Classification

- Schémas de classification
- Les environnements machines
- Caractéristiques de tâches
- Les critères d'optimisation

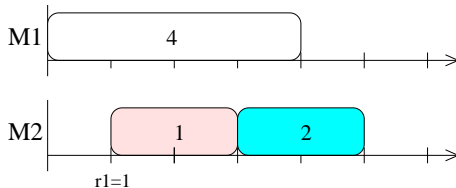
Les différentes classes d'ordonnement

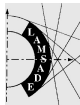
Complexité ...

- ... des algorithmes
- ... des problèmes

## Les différentes classes d'ordonnement

**Exemple** : 3 tâches, à exécuter sur deux machines en parallèle, telles que  $r_2 = r_3 = 0$ ,  $r_1 = 1$ ;  $p_1 = p_2 = 2$ ,  $p_3 = 4$





Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnement

Complexité ...

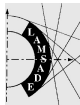
... des algorithmes  
... des problèmes

## Complexité des algorithmes

- Un même problème peut généralement être résolu par plusieurs algorithmes  
⇒ il faut comparer entre ces algorithmes
- La comparaison se base sur le **temps de calcul** et sur l'**espace mémoire** requis par l'algorithme

### Définition

On appelle **complexité en temps d'un algorithme dans le pire cas**, la fonction  $f(n)$  qui donne une borne supérieure du nombre d'opérations élémentaires effectuées par l'algorithme lorsque la taille de l'entrée est  $n$ .



Plan

Contexte

Classification

- Schémas de classification
- Les environnements machines
- Caractéristiques de tâches
- Les critères d'optimisation

- Les différentes classes d'ordonnement

Complexité ...

- ... des algorithmes
- ... des problèmes

## Complexité des algorithmes

### Définition

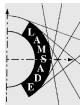
On dit que  $f(n) \in \mathcal{O}(g(n))$ , s'il existe une constante  $c > 0$  et un entier  $n_0$  tels que  $\forall n \geq n_0, |f(n)| \leq c|g(n)|$

### Définition

Un algorithme est de **complexité polynomiale** lorsque  $f(n) \in \mathcal{O}(p(n))$  et  $p$  est un polynôme en  $n$ , c'est-à-dire, il existe une constante  $k$  telle que  $f(n) \in \mathcal{O}(n^k)$

### Définition

Un algorithme dont la fonction complexité  $f(n)$  ne peut pas être majorée par un polynôme en  $n$  est dit **exponentiel**



Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnement

Complexité ...

... des algorithmes  
... des problèmes

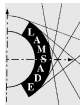
## Complexité des problèmes

Distinction entre problème de décision et problème d'optimisation

- Un **problème de décision** est un problème pour lequel une solution est soit "oui" soit "non"
- Un **problème d'optimisation** est un problème pour lequel on doit chercher à déterminer une solution qui optimise un critère
- A chaque problème d'optimisation on peut associer un problème de décision

### Définition

Un problème de décision appartient à la classe  $\mathcal{P}$ , s'il peut être résolu par un algorithme polynomial en  $n$



Plan

Contexte

Classification

- Schémas de classification
- Les environnements machines
- Caractéristiques de tâches
- Les critères d'optimisation

- Les différentes classes d'ordonnement

Complexité ...

- ... des algorithmes
- ... des problèmes

## Complexité des problèmes

### Définition

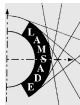
Un problème de décision appartient à la **classe  $\mathcal{NP}$**  s'il peut être résolu par un **algorithme non déterministe polynomial**.

### Définition

On considère deux problèmes de décisions  $Q$  et  $R$ . On dit que  **$Q$  se réduit polynomialement à  $R$** , et on note  $Q \propto R$ , s'il existe une fonction polynomiale  $g$  qui transforme toute instance de  $Q$  en une instance de  $R$  telle que  $x$  est une réponse "oui" de  $Q$  si et seulement si  $g(x)$  est une réponse "oui" de  $R$ .

### Définition

Un problème  $R$  est  **$\mathcal{NP}$ -complet** ssi  $R \in \mathcal{NP}$  et  $\forall Q \in \mathcal{NP}, \exists \propto$  telle que  $Q \propto R$ .



Plan

Contexte

Classification

Schémas de classification  
Les environnements machines  
Caractéristiques de tâches  
Les critères d'optimisation

Les différentes classes d'ordonnancement

Complexité ...

... des algorithmes  
... des problèmes

## Complexité des problèmes

- Un problème d'optimisation est dit  $\mathcal{NP}$ -difficile si le problème de décision associé est  $\mathcal{NP}$ -complet
- Il existe un certain nombre de résultats dans la littérature qui montrent les liens, sous forme d'**arbres de réduction**, entre différents problèmes d'ordonnancement.
- Arbre de réduction en fonction des **critères d'optimisation**

