Problèmes d'ordonnancement : Généralités et Problème central

Module MOP: Cours 1

Spécialité IAD

Philippe Chrétienne

LIVRES

- Problèmes d'ordonnancement: modélisation, complexité et algorithmes, Masson, 1988, J. Carlier et Ph. Chrétienne.
- Scheduling Algorithms, Peter Brucker, J. Wiley, 2004.
- Scheduling : Theory, Algorithms and Systems, M. Pinedo, 2003.
- Modèles et Algorithmes en Ordonnancement, GOTHA, 2004.

PLAN DU COURS

- Cadre des problèmes d'ordonnancement
- Problèmes sans contraintes de ressources.
- Problèmes à une machine.
- Problèmes à machines identiques.
- Problèmes d’ateliers.
- Ordonnancement de projets (RCPSP)

ENVIRONNEMENT

- Un ensemble de tâches $J_1, J_2, \ldots.., J_n$ à exécuter.
- Un mode d'exécution des tâches

  (mode préemptif, mode non préemptif, …)
- Des ressources de type $R_1, \ldots, R_q$ en quantité limitée $r_1, \ldots., r_q$.
- Des contraintes temporelles entre les tâches.
- Des besoins en ressources $R_1, \ldots, R_q$ nécessaires à l'exécution des tâches :

  exemple : b_{ik} quantité de ressource de type k requise pendant toute l'exécution de J_i.

Ordonnancement (réalisable) :
Allouer à chaque tâche :

- un intervalle d'exécution (si mode non préemptif) ou plusieurs (si mode préemptif);
- des ressources ;

de telle sorte que :

- les contraintes temporelles soient satisfaites ;
- les contraintes de ressources soient satisfaites.

Evaluation (monocritère) d'un ordonnancement :
Un critère associe à chaque ordonnancement un coût.
Exemples : délai total, plus grand retard, …

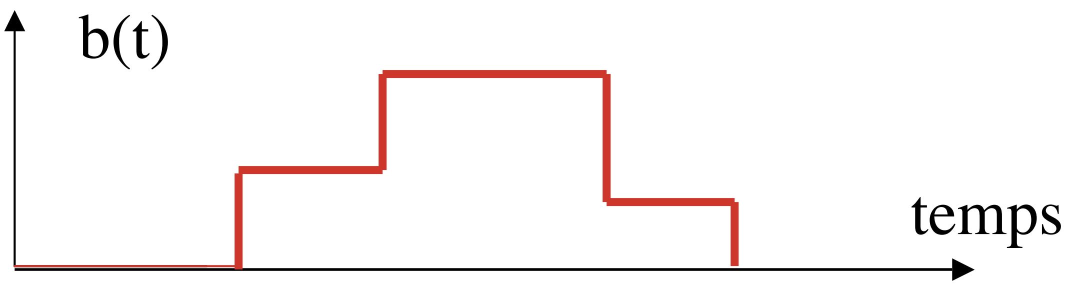
Problème d'ordonnancement (monocritère) :
Déterminer un ordonnancement de coût minimal.

Ressources

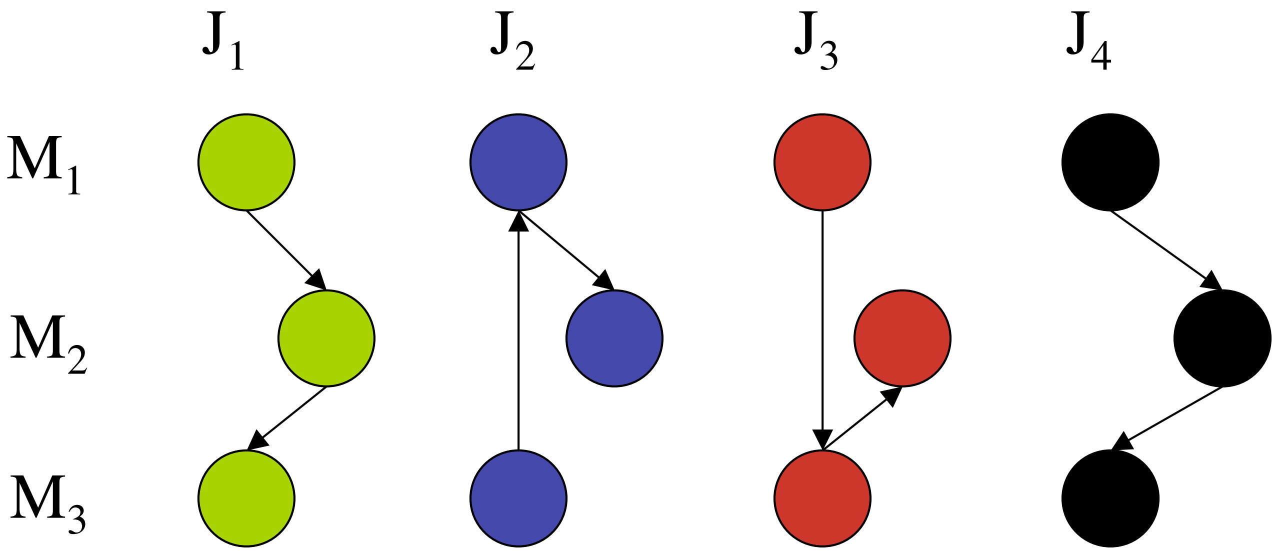
Ressource renouvelable :
A nouveau disponible après son utilisation par une tâche
(ex : une machine, un fichier, …)

Ressource consommable :
Consommée par la tâche qui l'utilise (ex: argent, énergie,…)

Courbe de disponibilité d'une ressource :

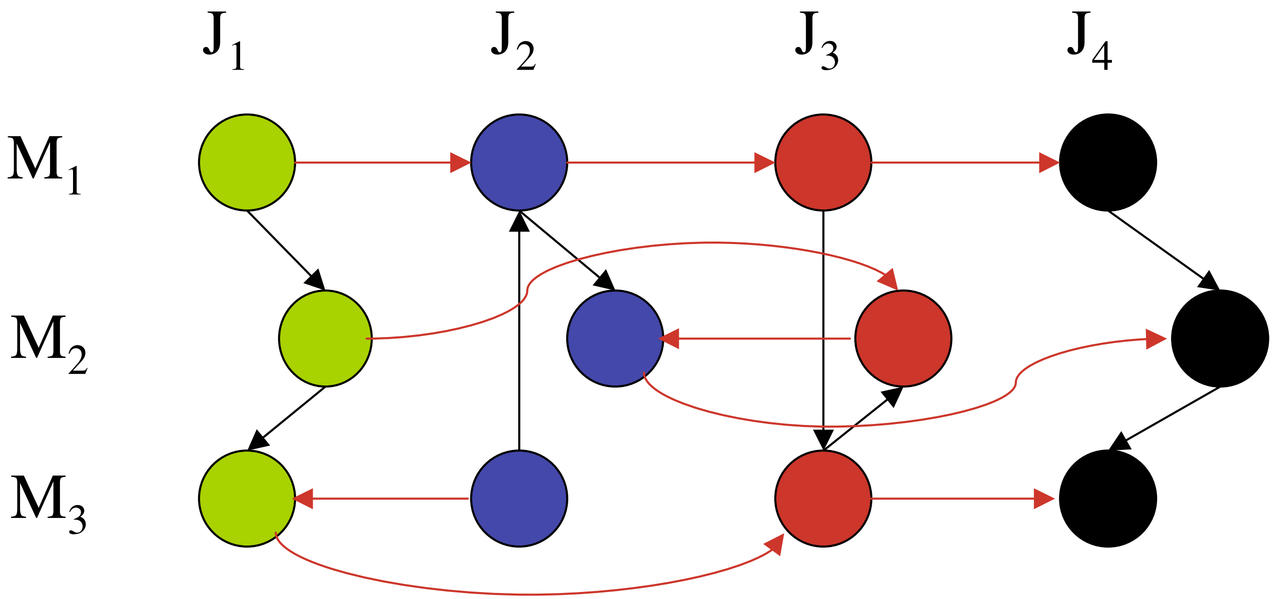


Ressource spécialisée :
Ne peut être allouée qu'à certaines tâches.
Exemple :
- réalisation de N pièces ;
- chaque pièce doit être usinée par m machines distinctes
  dans un certain ordre fixé a priori.
La tâche (opération) T_{ik} correspond à l'usinage de la pièce i par la machine k.
C'est le problème du job shop (version standard).



Exemple:
4 pièces
3 machines

Il faut séquencer sur chaque machine k les 4 opérations T_{ik}.

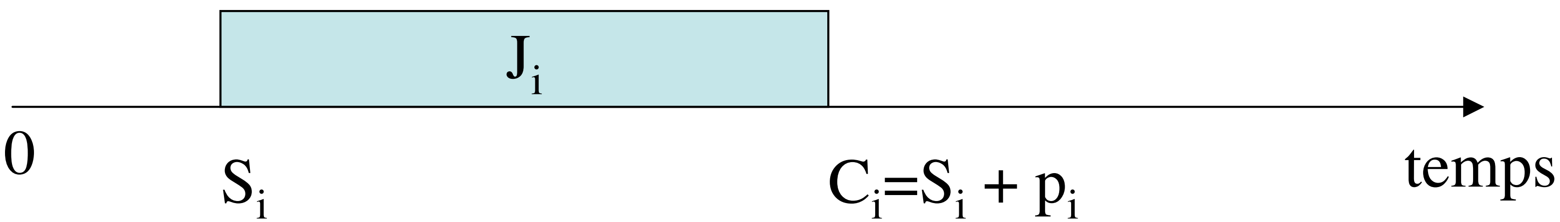


Exemple de séquencement réalisable (pas de circuit).

Contraintes temporelles

Une tâche J_i a une durée.
Si la durée de J_i ne dépend pas des ressources qu'elle utilise, on la note p_i.

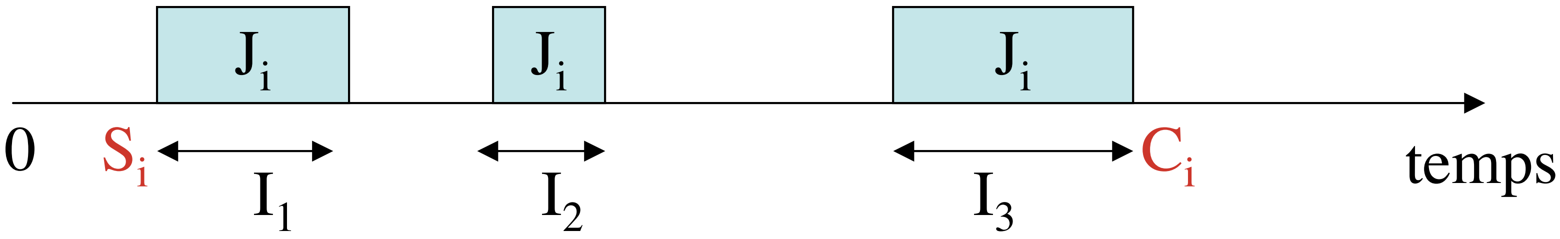
Intervalle(s) d'exécution de la tâche J_i .
En mode non préemptif, un seul intervalle par tâche :



S_i : date de début ;
C_i : date de fin.

En mode préemptif, l'exécution d'une tâche peut être répartie sur plusieurs intervalles de temps.

Intervalles d'exécution d'une tâche préemptive.



$$l(I_1) + l(I_2) + l(I_3) = p_i$$

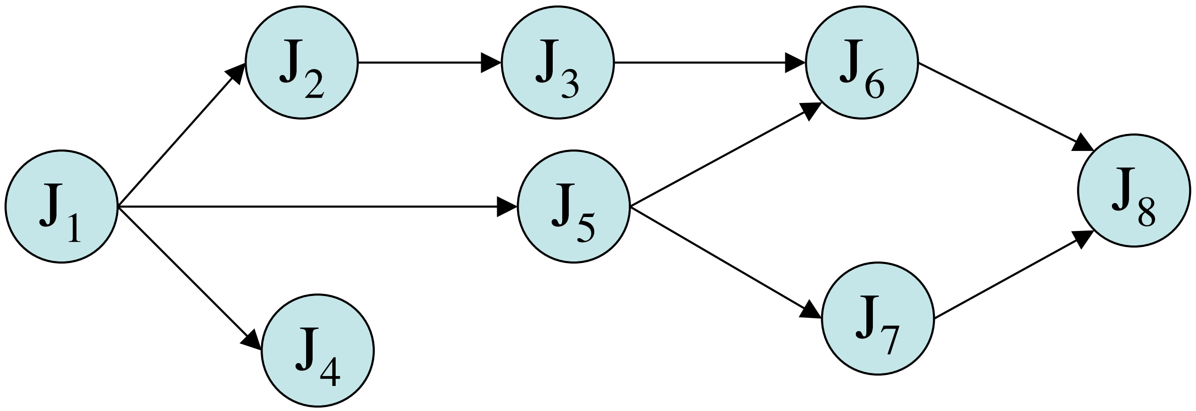
$$C_i \geq S_i + p_i$$

Contraintes de précédence

Précédence simple : $(J_i,J_j) \Leftrightarrow S_j \geq C_i$

Précédence généralisée : $(J_i,J_j,a_{ij}) \Leftrightarrow S_j \geq C_i + a_{ij}$.

Graphe de précédence :
Sommets : tâches ;
Arcs : couples de précédence simple.



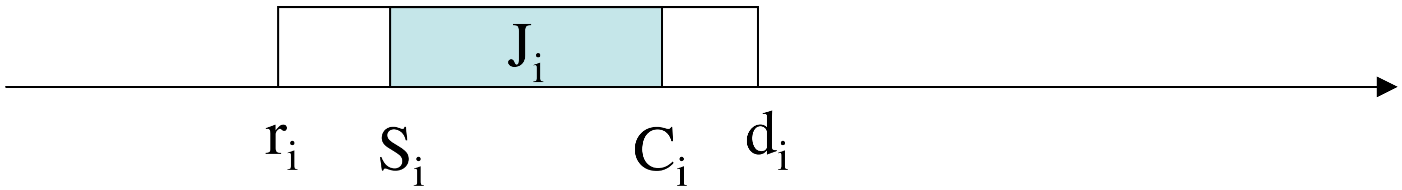
Contraintes de fenêtre

Date de disponibilité r_i de la tâche J_i

Contrainte : $S_i \geq r_i$.

Deadline « dur » d_i de la tâche J_i

Contrainte : $C_i \leq d_i$.



Si la tâche J_i est soumise à une date de disponibilité et un deadline dur, alors $[r_i,d_i]$ est la fenêtre d'exécution de la tâche J_i

Deadline « mou » Δ_i de la tâche J_i (date de fin attendue, mais non imposée)

Critères

Délai total (makespan): $C_{max} = \max_i C_i$;
Délai moyen (flowtime) : $(1/n)\sum C_i$;
Délai moyen pondéré (weighted flowtime) : $\sum w_i C_i$;

Plus grand retard (maximum tardiness) : $\text{Max } T_i$ où $T_i = \max\{0, C_i - d_i\}$
Somme pondérée des retards : $\sum w_i T_i$;

Plus grand retard algébrique : $\text{Max } L_i$ où $L_i = C_i - d_i$.

Nombre de tâches en retard : $\sum U_i$ où $U_i = 1$ si $C_i > d_i$, 0 sinon ;
Nombre pondéré de tâches en retard : $\sum w_i U_i$

Coût d'avance et retard : $\sum \alpha_i E_i + \beta_i T_i$ où $E_i = \max\{0, d_i - C_i\}$

Critère régulier :

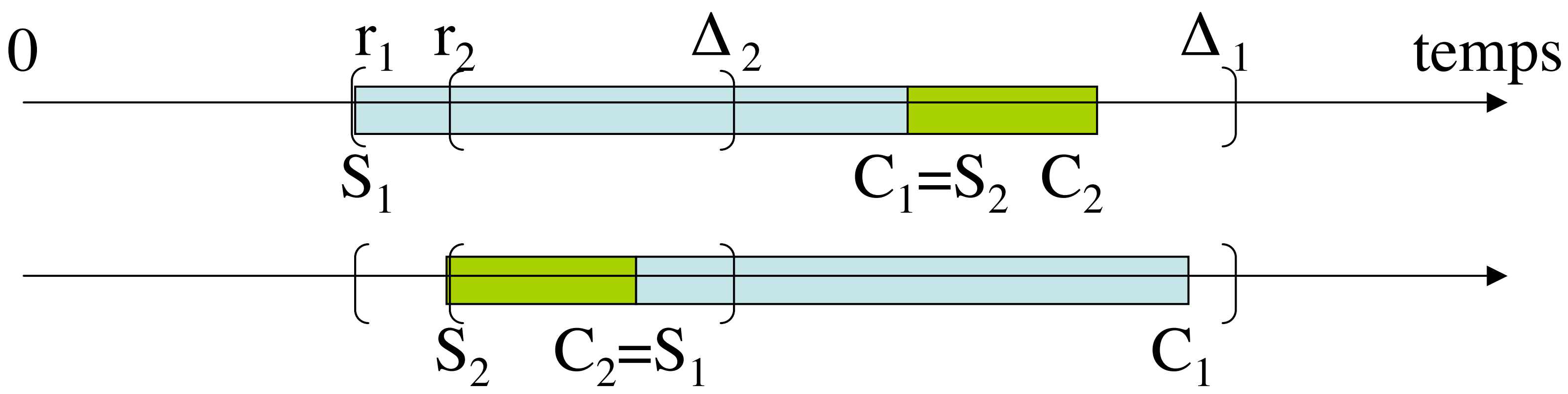
Fonction $\Phi(C_1,..,C_n)$ croissante au sens large :

$$(C'_1,..,C'_n) \geq (C_1,..,C_n) \Rightarrow \Phi(C'_1,..,C'_n) \geq \Phi(C_1,..,C_n)$$

Les critères réguliers ont été majoritairement étudiés.
On cherche à exécuter les tâches au plus tôt.

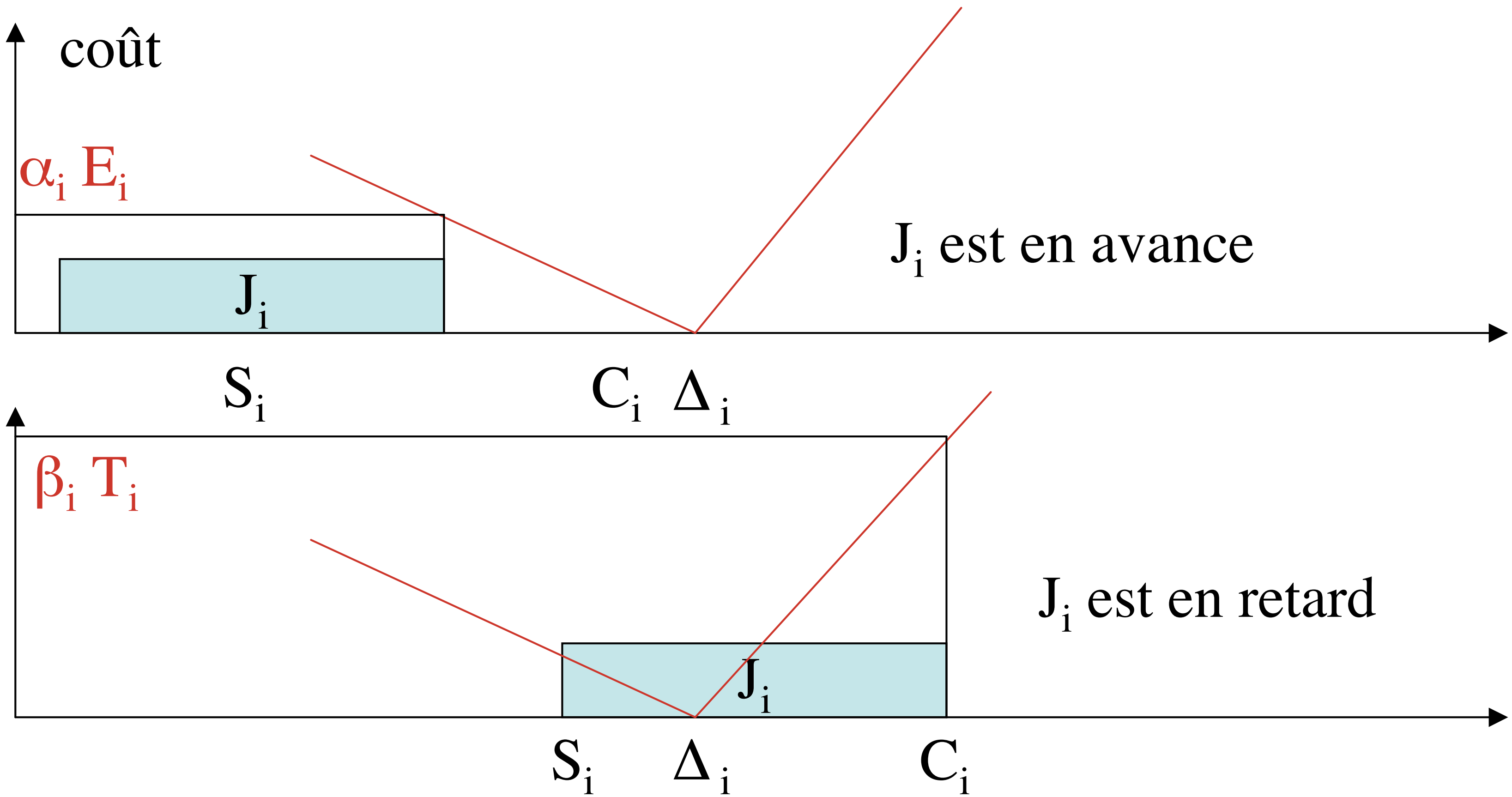
Cependant il peut être intéressant d'introduire des temps morts pour attendre que des tâches plus prioritaires soient exécutables.

Exemple : 2 tâches J_1 (bleu) et J_2 (vert) , dates de disponibilité r_1 et r_2 , dates échues Δ_1 et Δ_2.



Dans les systèmes de production, terminer une tâche (fabrication d'un équipement) avant sa date de fin attendue (date de livraison) induit un stockage de l'équipement.
Intérêt de la production juste à temps.

Exemple : coûts d'avance et de retard linéaires



Typologie $\alpha \mid \beta \mid \gamma$

Champ α : caractéristiques des ressources.

Exemples :

- P3 (3 machines identiques)
- P (m machines identiques)
- R (m machines distinctes)
- F2 (flow shop à 2 machines)
- …..

Champ γ : caractérisriques des critères.

Exemples :

- C_{max} (makespan)
- $\sum T_i$ (somme des retards)
- $\sum \alpha_i E_i + \beta_i T_i$ (somme des coûts d'avance-retard)

Champ β : caractéristiques des tâches.

Exemples :

- pmtn (préemption autorisée)
- r_i (dates de disponibilité)
- $p_i=1$ (durées unitaires)
- prec (graphe des précédences quelconque)

…….

Exemple :

Problème $P3 \mid p_i=1, prec \mid C_{max}$

- 3 machines identiques ;
- durées unitaires ;
- graphe des précédences quelconque ;
- minimiser le makespan

Le problème central de l’ordonnancement

- Définition
- Ensemble de potentiels d'un graphe valué
- Ordonnancement au plus tôt
- Ordonnancement au plus tard
- Exemple

Définition du PCO

Tâches non préemptives $J_1, J_2, \ldots, J_n$;
La durée de J_i est notée p_i.
Contraintes de précédence généralisées :

m triplets (J_i,J_j,a_{ij}) ;

$(J_i,J_j,a_{ij}) \Leftrightarrow S_j - S_i \geq a_{ij}$;

Aucune contrainte de ressource ;
Critère : C_{max} .

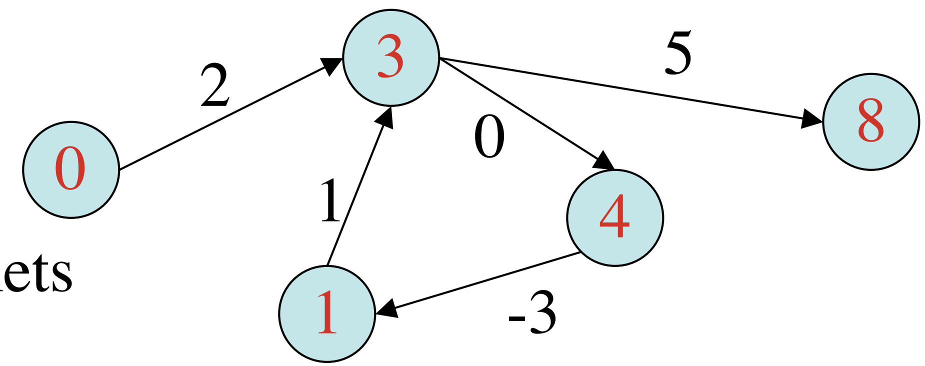
Remarque :
Le triplet (J_i,J_j,p_i) correspond à la contrainte de succession simple entre J_i et J_j .

Ensemble de potentiels

Soit $G = (S,A)$ un graphe
Soit v une valuation des arcs de G.
$S = \{s_1,\dots,s_n\}$: ensemble des sommets ;
$A = \{a_1,\dots,a_m\}$: ensemble des arcs ;
$v : A \rightarrow \Re$ une valuation réelle des arcs.

$t : S \rightarrow \Re$ est un ensemble de potentiels (edp) de (G,v) si :
$\forall\, a \in A,\ t(a^+) - t(a^-) \geq v(a)$.

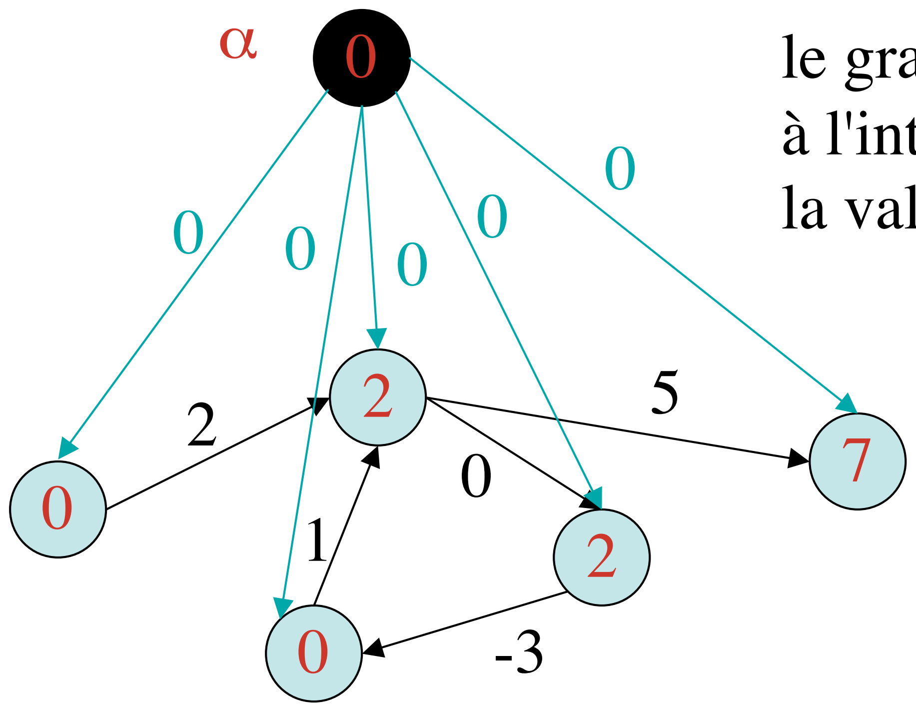
Exemple :
v en noir sur les arcs
t en rouge dans les sommets



On note $T(G,v)$ l'ensemble des edp de G pour la valuation v.

Propriété 1:
$T(G,v) = \varnothing$ si et seulement si G possède un circuit de valuation strictement positive.

Preuve :
Condition suffisante:
En sommant les inégalités de potentiel associées aux arcs d'un circuit c, on obtient $v(c) \leq 0$.
Condition nécessaire :
Supposons que G ne possède pas de circuit « strictement positif ».
Ajoutons à G un sommet α et, pour tout sommet s de S, un arc (α,s) de valuation nulle.
Notons G' le nouveau graphe.



le graphe G'
à l'intérieur du sommet s :
la valeur l(α,s).

Tout circuit de G' est de valeur négative ou nulle.
Donc, dans G', les chemins de valeur maximum de α à s existent.
Soit l(α,s) la valeur maximum d'un chemin de α à s dans G'.
On a : $\forall\ a \in A,\ l(\alpha, a^{+}) - l(\alpha, a^{-}) \geq v(a)$.
Donc $T(G,v) \neq \emptyset$.
Remarque : On a aussi : $\forall\ s \in S\ ,\ l(\alpha,s) \geq 0$.

Autres propriétés de T(G,v) :

Propriété 2 :

Si $t \in T(G,v)$ alors $\forall r \in \Re$, $t + r(1,\ldots,1) \in T(G,v)$

Propriété 3 :

Si $T(G,v) \neq \emptyset$, alors T(G,v) possède un potentiel $t \geq 0$

Propriété 4 :

T(G,v) est convexe.

(solutions d'un système fini d'inégalités linéaires)

Propriété 5 :

Si $f \in T(G,v)$ et $g \in T(G,v)$

alors $x = \mathrm{Min}\{f,g\} \in T(G,v)$ et $y = \mathrm{Max}\{f,g\} \in T(G,v)$

où : $x_i = \mathrm{Min}\{f_i,g_i\}$ et $y_i = \mathrm{Max}\{f_i,g_i\}$

Propriété 6 :

Soit $T^+(G,v)$ l'ensemble des edp positifs ou nuls de G.

$T^+(G,v)$ possède un plus petit élément e défini par :

$\forall\ s \in S$, $e(s) = l(\alpha,s)$.

Preuve :

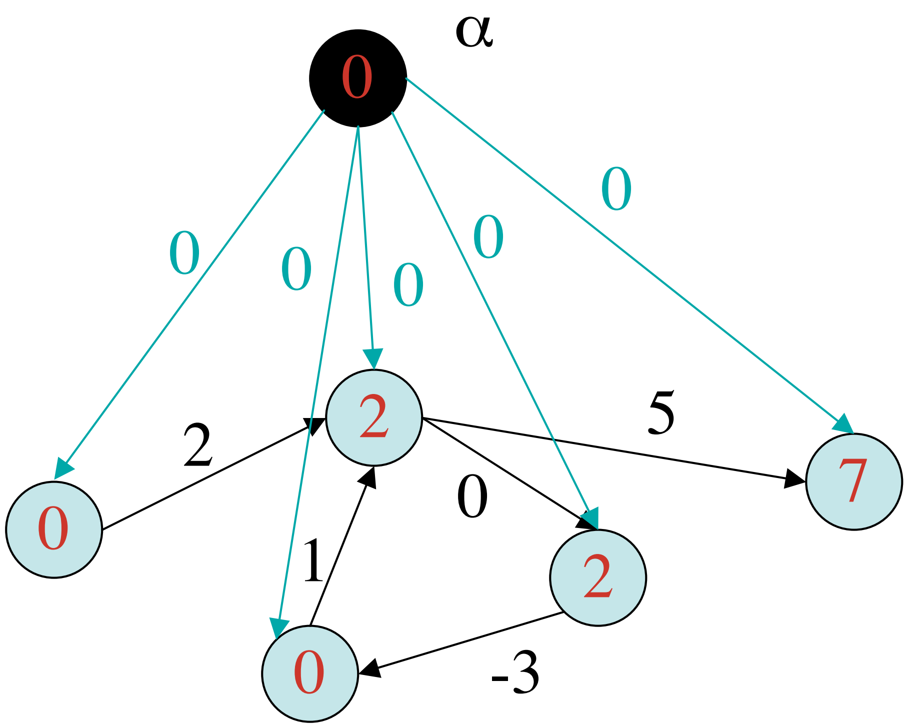
Soit $t \in T^+(G,v)$.

Soit c un chemin de valeur maximum de α à s dans G'.

Soit a le premier arc de c et soit c' le sous-chemin de c de u à s

On a : $t(s) - t(u) \geq v(c') = v(c) = l(\alpha,s)$.

Il en résulte que : $t(s) \geq l(\alpha,s)$.



Le plus petit élément de $T^+(G,v)$.

Retour sur le PCO

Tâches non préemptives $J_1, J_2, \ldots, J_n$;
Contraintes de précédence généralisées :
• m triplets (J_i,J_j,a_{ij}) ;
• $(J_i,J_j,a_{ij}) \Leftrightarrow S_j - S_i \geq a_{ij}$;
Critère : C_{max} .

Soit G le graphe des précédences.
Propriété :
Les ordonnancements sont en bijection avec les edp de $T^+(G,a)$.

Propriété :
Il existe un ordonnancement si et seulement si G n'a pas de circuit de valeur strictement positive.

Ordonnancements au plus tôt et au plus tard

Hypothèse : T(G,a) non vide.

Le plus petit élément de $T^+(G,a)$ est appelé ordonnancement au plus tôt.

Ajoutons à G :

- 2 sommets J_0 et J_{n+1} ;
- pour chaque tâche J_i,
  - un arc (J_0,J_i) de valuation nulle ;
  - un arc (J_i,J_{n+1}) de valuation p_i .

Notons G'' et a'' le graphe ainsi obtenu et sa valuation

Notons $T^*(G'',a'')$ l'ensemble des edp de G'' pour a''
tels que $t_{n+1} = l(J_0,J_{n+1})$.

Propriété 7 :
Soit $u : S \rightarrow \Re$ défini par $u_i = l(J_0,J_{n+1}) - l(J_i,J_{n+1})$.
u est le plus grand élément de $T^*(G'',a'')$.

Preuve :

$l(J_0,J_{n+1}) \geq l(J_0,J_i) + l(J_i,J_{n+1})$ et donc $u_i \geq 0$.

$u_{n+1} = l(J_0,J_{n+1})$.

Soit t un edp de T*(G'',a'').

Soit c un chemin de J_i à J_{n+1} de valeur $l(J_i,J_{n+1})$.

On a : $t_{n+1}-t_i \geq l(J_i,J_{n+1})$ et $t_{n+1} = l(J_0,J_{n+1})$.

Il en résulte : $u_i = l(J_0,J_{n+1}) - l(J_i,J_{n+1}) \geq t_{n+1}- l(J_i,J_{n+1}) \geq t_i$

Soit (J_i,J_j) un arc de G.

On a : $u_j - u_i = l(J_i,J_{n+1})-l(J_j,J_{n+1}) \geq a_{ij}$;

Pour un arc (J_0,J_i), on a : $u_i - u_0 \geq 0$;

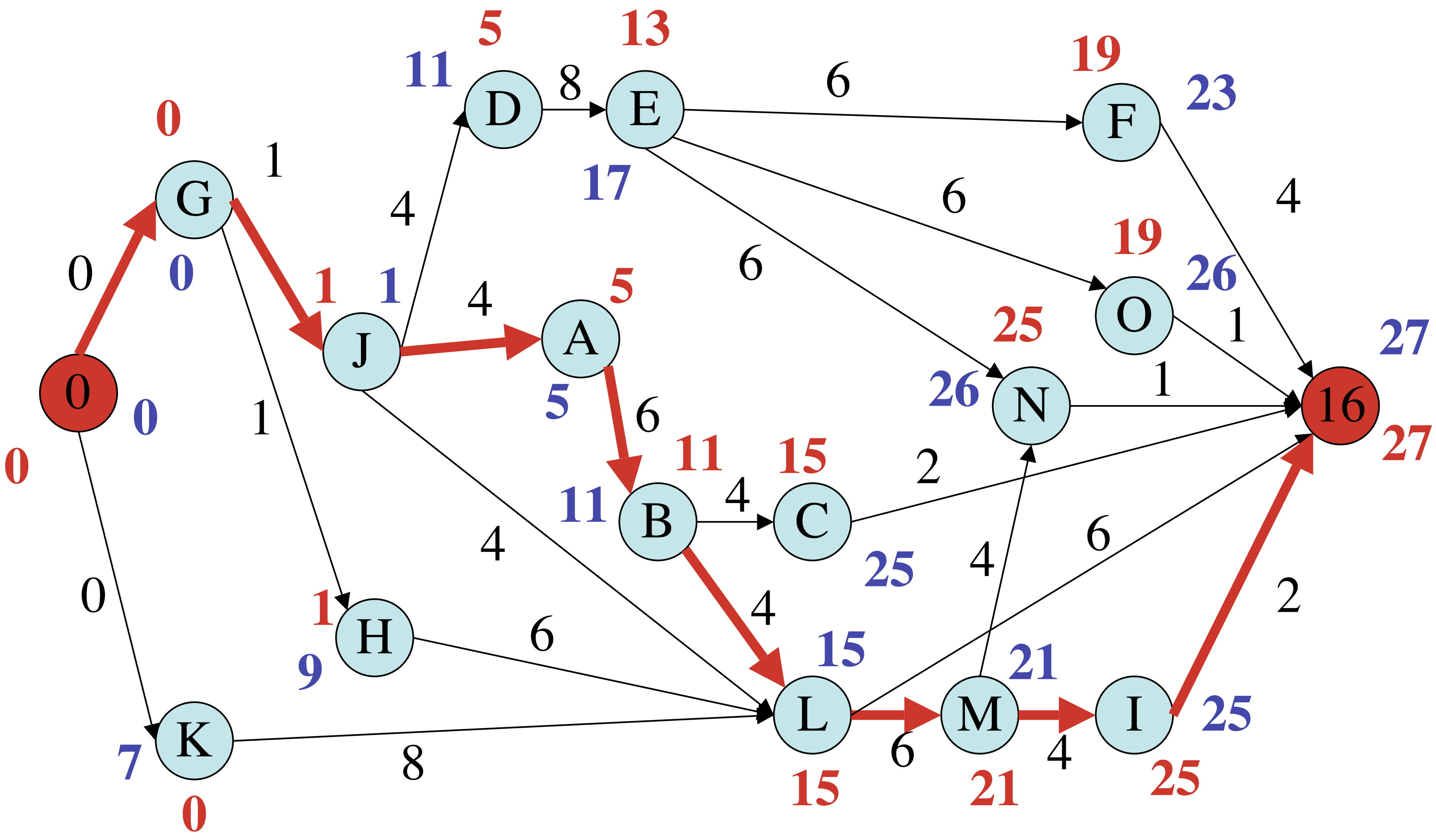
Pour un arc (J_i,J_{n+1}) on a $u_{n+1} - u_i = l(J_i,J_{n+1}) \geq p_i$.

u est donc le plus grand edp de T*(G '',a'').

u est aussi appelé : « ordonnancement au plus tard de (J,G,a) »

Exemple

J_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
p_i	6	4	2	8	6	4	1	6	2	4	8	6	4	1	1
P_i	J	A	B	J	D	E	$\varnothing$	G	M	G	$\varnothing$	B J H K	L	E M	E J



Graphe des précédences,
Dates au plus tôt (en rouge),
Dates au plus tard (en bleu),
Chemin critique (arcs rouges)