

## Ordonnancement de projets

### Ordonnancement de projets

La planification d'un projet regroupe l'ensemble des méthodes permettant de trouver l'organisation optimale du projet (durée minimale, identification des tâches critiques, ...). Ce problème peut être modélisé à l'aide d'un réseau PERT ("project evaluation and review technique") et résolu à l'aide de la théorie des graphes.

#### Définition (Projet)

Un projet  $\mathcal{P}$  est constitué par :

- ▶ un ensemble de tâches à réaliser  $(A_i)_{i=1, \dots, n}$  avec, pour chaque tâche, une date de commencement  $t_i$  et une durée  $d_i$  ;
- ▶ un ensemble de "contraintes" sur les tâches du projets, contraintes qui peuvent s'exprimer par des inégalités faisant intervenir les dates  $(t_i)_{i=1, \dots, n}$  et les durées  $(d_i)_{i=1, \dots, n}$ .

Trouver un ordonnancement pour le projet consiste à calculer, à partir des durées  $(d_i)_{i=1, \dots, n}$ , des dates de commencement  $(t_i)_{i=1, \dots, n}$  qui soient compatibles avec les contraintes du projet.

## Ordonnancement de projets

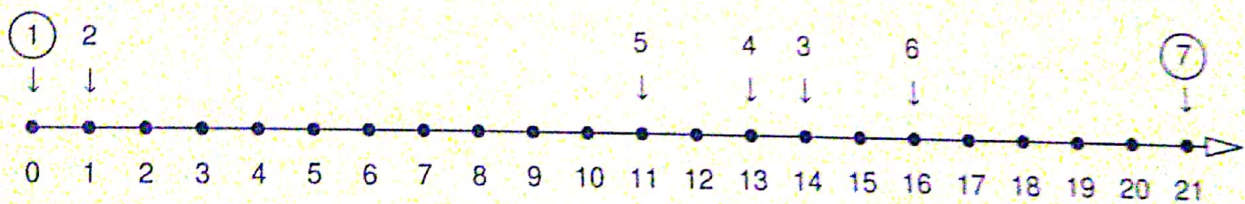
Un projet et ses contraintes se présentent en général sous forme de tableau :

Tâches	Opérations et contraintes	Durée (jours)
1	début du projet	0
2	aucune contrainte	11
3	commence au plus tôt 1 jour après le début du projet commence au plus tard 8 jours après le début de (4)	5
4	commence au plus tôt 1 jour après la fin de (2)	8
5	commence au plus tôt 1 jour avant la fin de (2)	5
6	commence au plus tôt après le début de (4) commence au plus tôt après la fin de (5)	4
7	fin du projet	0

**Concevoir par l'intuition un planning directement à partir d'une liste de contraintes est quasi-impossible, en pratique, surtout si le nombre de tâches à effectuer est important.**

## Ordonnancement de projets

Par exemple, l'ordonnancement suivant est compatible :



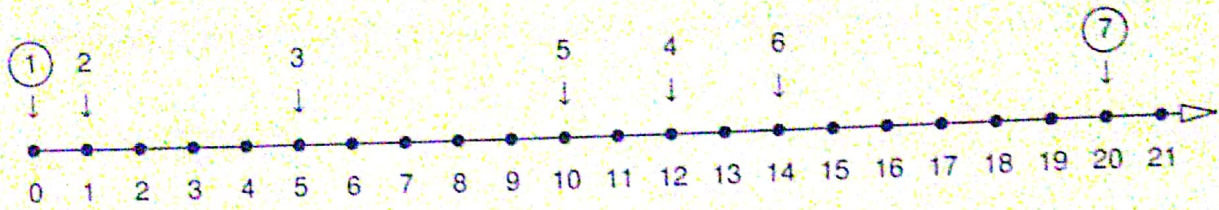
Cet ordonnancement s'écrit sous la forme compacte :

Tâche	1	2	3	4	5	6	7
Date de début	0	1	14	13	11	16	21

Il est compatible car toutes les contraintes présentées dans le cahier des charges sont vérifiées.

## Ordonnancement de projets

En revanche, l'ordonnancement suivant n'est **pas** compatible :



Il s'écrit sous la forme compacte :

Tâche	1	2	3	4	5	6	7
Date de début	0	1	5	12	10	14	20

## Ordonnancement de projets | Contraintes

Pour rendre tout cela plus lisible nous allons traduire les contraintes sous forme d'inégalités entre les dates de commencement de chaque tâche. Il faudra pour cela bien identifier les différents types de contrainte.

### Proposition (Type de contraintes)

On traduira les différentes contraintes en inégalités de la manière suivante :

- ▶ *contrainte au plus tôt* : "la tâche  $A_j$  commence au plus tôt  $\delta$  après le début de la tâche  $A_i$ " :
- ▶ *contrainte au plus tard* : "la tâche  $A_j$  commence au plus tard  $\delta$  après le début de la tâche  $A_i$ " :
- ▶ *contraintes implicites* : "toute tâche  $A_i$  doit démarrer au plus tôt au début du projet ( $A_1$ ) et finir au plus tard à la fin du projet ( $A_n$ )" :

Pour l'exemple considéré, traduction de chaque contrainte par une inégalité :

Contraintes	Inéquation
(3) commence au plus tôt 1 jour après le début de (1)	$t_3 - t_1 \geq 1$
(3) commence au plus tard 8 jours après le début de (4)	$t_3 - t_4 \leq 8$
(4) commence au plus tôt 1 jour après la fin de (2)	$t_4 - t_2 \geq 1$
(5) commence au plus tôt 1 jour avant la fin de (2)	$t_5 - t_2 \geq -1$
(6) commence au plus tôt après le début de (4)	$t_6 - t_4 \geq 0$
(6) commence au plus tôt après la fin de (5)	$t_6 - t_5 \geq 5$

## Ordonnancement de projets | Graphe associé aux contraintes

À partir des équations obtenues on va pouvoir représenter ce projet par un graphe orienté et valué.

### Définition (Graphe potentiel-tâches)

On associe à un problème d'ordonnancement un graphe orienté et valué  $G = (S, A, f)$  tel que :

- chaque tâche  $A_i$  du projet est représentée par un sommet  $x_i$  du graphe ;
- chaque contrainte  $t_j - t_i \geq d$  du projet est représentée par un arc  $(x_i, x_j)$  de valuation  $f(x_i, x_j) = d$ .

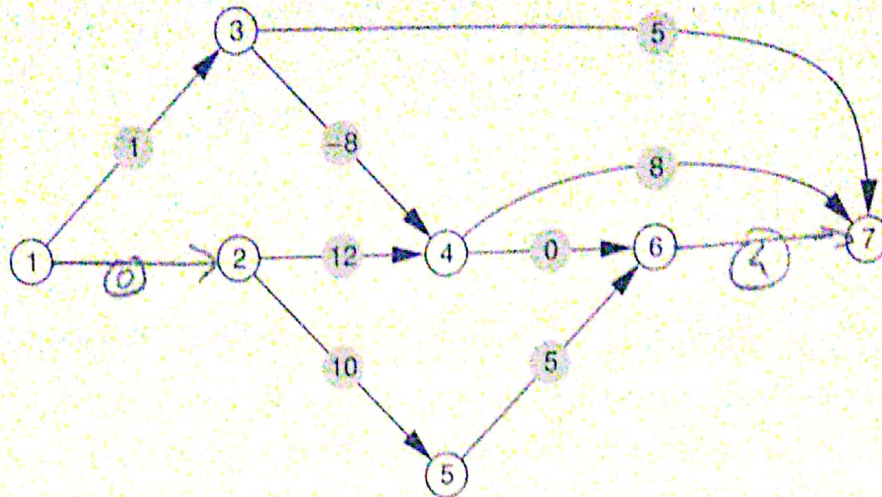
On ajoute éventuellement des **contraintes implicites** pour que chaque tâche dans le graphe appartienne à au moins un chemin reliant la tâche de début de projet et celle de fin de projet.



Il est **impératif** de modéliser le début de projet et la fin de projet par des tâches  $A_1$  et  $A_n$  de durée nulle.

## Ordonnancement de projets | Graphe associé aux contraintes

Pour l'exemple considéré, le graphe associé au projet est le suivant :



## Ordonnancement de projets | Ordonnancement au plus tôt

**Théorème (Ordonnancement au plus tôt)**

L'ordonnancement au plus tôt d'un projet consiste à trouver les dates de commencement  $\{t_i^{(-)}\}_{i=1,\dots,n}$  de chaque tâche telles que le projet soit fini le plus rapidement possible.

Pour déterminer l'ordonnancement au plus tôt d'un projet, il suffit de déterminer les chemins les plus longs dans le graphe  $G$  depuis le sommet correspondant au début du projet. Les distances  $\{d_i^{(-)}\}_{i=1,\dots,n}$  ainsi obtenues sont les dates de commencement au plus tôt de chaque tâche :

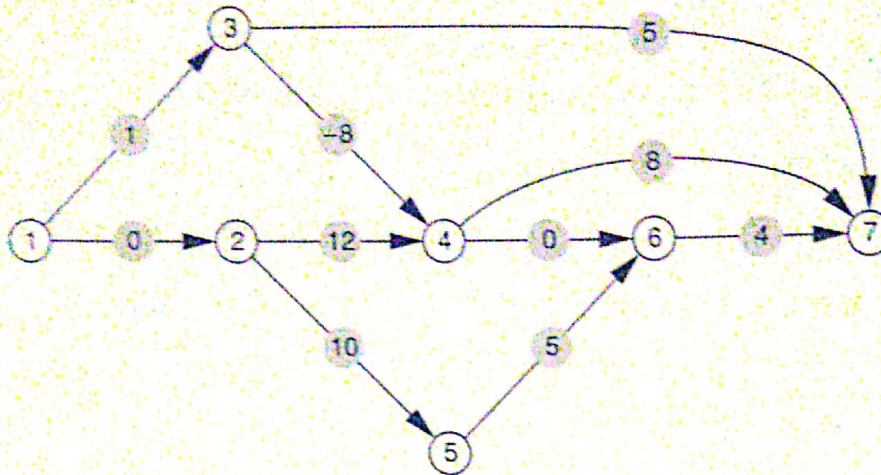
$$t_i^{(-)} = d_i^{(-)}, \quad \forall i = 1, \dots, n.$$

La durée totale minimale du projet est  $T_{\min} = t_n^{(-)}$ .

**Remarque.** Lorsqu'il n'y a pas de circuits dans le graphe on peut utiliser l'algorithme de Bellman pour calculer l'ordonnancement correspondant.  $\square$

## Ordonnancement de projets | Ordonnancement au plus tôt

Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



Le graphe  $G$  ne possède pas de circuits ; on peut donc utiliser l'algorithme de Bellman (le graphe est déjà décomposé en niveaux) :

$i$	1	2	3	4	5	6	7
$d(i) (P(i))$	0, 0	0, 1	1, 1	12, 2	10, 7	15, 5	20, 4

## Ordonnancement de projets | Ordonnancement au plus tôt

Les dates de début de tâche  $t_j^{(-)}$  sont les "distances"  $d(i)$  obtenues par la mise en œuvre de l'algorithme de Bellman avec recherche de plus longs chemins.

Les dates de début de tâche sont donc les suivantes :

$i$	1	2	3	4	5	6	7
$t_j^{(-)} = d(i)$	0	0	1	12	10	15	20

La durée minimale de réalisation du projet est donc de

$$T_{\min} = 20$$

## Ordonnancement de projets | Ordonnancement au plus tard

### Definition (Graphe réciproque)

Soit  $G = (S, A, f)$  un graphe orienté valué. Le **graphe réciproque de  $G$**  désigne le graphe  $G^{-1} := (S, B, g)$  tel que

$$B = \{(y, x) \text{ tel que } (x, y) \in A \text{ et } \forall (x, y) \in B, g(x, y) = f(y, x)\}.$$

### Théorème (Ordonnancement au plus tard)

Pour une durée  $T \geq T_{\min}$ , l'**ordonnancement au plus tard d'un projet** consiste à trouver les dates les plus tardives de commencement  $\{t_i^{(+)}\}_{i=1, \dots, n}$  de chaque tâche telles que la durée du projet soit au maximum de  $T$ .

Pour déterminer l'ordonnancement au plus tard d'un projet, il suffit de déterminer les **chemins les plus longs dans le graphe réciproque  $G^{-1}$**  depuis le sommet correspondant à la fin du projet. Les distances  $\{d_i^{(+)}\}_{i=1, \dots, n}$  ainsi obtenues sont le temps restant à partir du commencement d'une tâche jusqu'à la fin du projet; en d'autres termes la tâche  $i$  doit démarrer au plus tard à la date :

$$t_i^{(+)} := T - d_i^{(+)}, \quad \forall i = 1, \dots, n.$$

## Ordonnancement de projets | Ordonnancement au plus tard

**Remarque.** Si le graphe  $G$  n'a pas de circuit, alors il en va de même pour le graphe réciproque  $G^{-1}$ . Les deux graphes peuvent être décomposés en niveaux et on peut donc utiliser l'algorithme de Bellman pour calculer les ordonnancements. □

Déterminons l'ordonnancement au plus tard pour  $T = 22$  jours.

**Étape 1.** On détermine d'abord le graphe réciproque de  $G$  (depuis le sommet correspondant à la fin du projet).

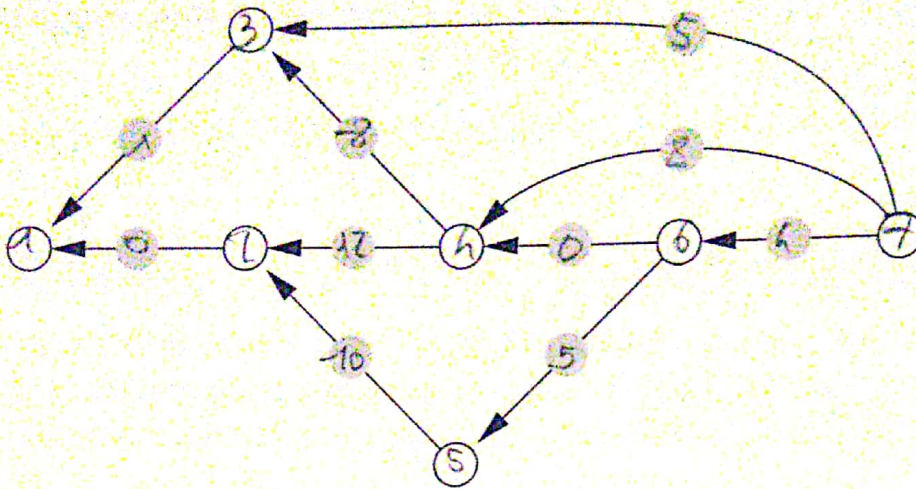
**Étape 2.** On applique l'algorithme de Bellman sur le graphe réciproque.

**Étape 3.** On détermine les dates d'ordonnancement au plus tard pour un temps maximal  $T = 22$  jours à partir des "distances"  $d(i)$  obtenues par la mise en œuvre de l'algorithme de Bellman sur le graphe réciproque avec recherche de plus longs chemins. Les dates d'ordonnancement au plus tard, notées  $t_i^{(+)}$ , sont alors définies par :

$$t_i^{(+)} := T - d(i), \quad \forall i = 1, \dots, n.$$

## Ordonnancement de projets | Ordonnancement au plus tard

Étape 1. On détermine d'abord le graphe réciproque de G.



Étape 2. On applique l'algorithme de Bellman sur le graphe réciproque à partir du sommet correspondant à la fin du projet (**Attention** : il faut, au préalable, effectuer un tri topologique du graphe réciproque).

$i$	7	6	4	5	2	3	1
$d(i) (P(i))$	0	4,7	8,7	9,6	20,4	5,7	20,8

## Ordonnancement de projets | Ordonnancement au plus tard

Étape 3. On détermine les dates d'ordonnancement au plus tard pour un temps maximal  $T = 22$  jours à partir des "distances"  $d(i)$  obtenues par la mise en œuvre de l'algorithme de Bellman sur le graphe réciproque avec recherche de plus longs chemins :

$i$	1	2	3	4	5	6	7
$d(i) (P(i))$	20(2)	20(4)	5(7)	8(7)	9(6)	4(7)	0(+∞)

Les dates d'ordonnancement au plus tard  $t_i^{(+)}$  sont définies par :

$$t_i^{(+)} := T - d(i), \quad \forall i = 1, \dots, n.$$

On obtient donc :

$i$	1	2	3	4	5	6	7
$t_i^{(+)}$	2	2	17	14	13	18	22

L'ordonnancement au plus tard à 22 jours du projet permet d'envisager les débuts de tâches aux dates indiquées dans le tableau ci-dessus.

## Ordonnancement de projets | Tâches critiques

L'ordonnancement au plus tôt permet de visualiser les **tâches critiques** du projet, pour lesquelles on ne peut pas prendre de retard sans répercussion sur la date de fin du projet.

### Théorème (Chemin critique)

Si  $T_{min}$  est la durée minimale du projet (calculée via l'ordonnancement au plus tôt), un chemin le plus long depuis la tâche de début de projet jusqu'à celle de fin de projet est appelé **chemin critique**. Tout retard sur la réalisation d'une des tâches de ce chemin critique (que l'on désigne comme une **tâche critique**) entraîne un allongement de la durée du projet.

**Remarque.** Un chemin critique s'obtient grâce à la liste des "prédécesseurs (ou pères)" obtenue à l'issue de la mise en œuvre de l'algorithme de Bellman pour l'ordonnancement au plus tôt. □

**Remarque.** Pour déterminer **tous** les chemins critiques, il faut donc avoir déterminé, au préalable, **tous** les plus longs chemins depuis la tâche de début de projet jusqu'à celle de fin de projet. □

## Ordonnancement de projets | Tâches critiques

Pour notre projet, la liste des "prédécesseurs (ou pères)" déterminée avec l'ordonnancement au plus tôt est la suivante (voir précédemment) :

$i$	1	2	3	4	5	6	7
$P(i)$	0	1	1	2	2	5	4

En partant de la fin, on a alors

On en déduit que

