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.

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

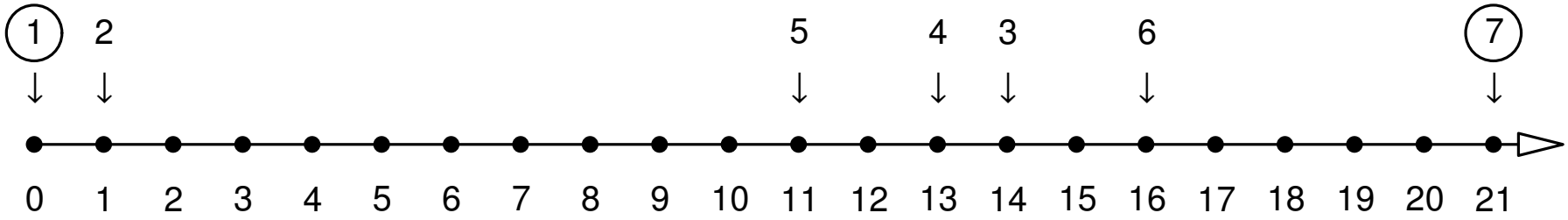
Tâches	Opérations et contraintes	Durée (jours)
1	*début du projet*	0
2	aucune contrainte	11
3	commence au plus tôt 1 jour après le début du projet commence au plus tard 8 jours après le début de (4)	5
4	commence au plus tôt 1 jour après la fin de (2)	8
5	commence au plus tôt 1 jour avant la fin de (2)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

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.

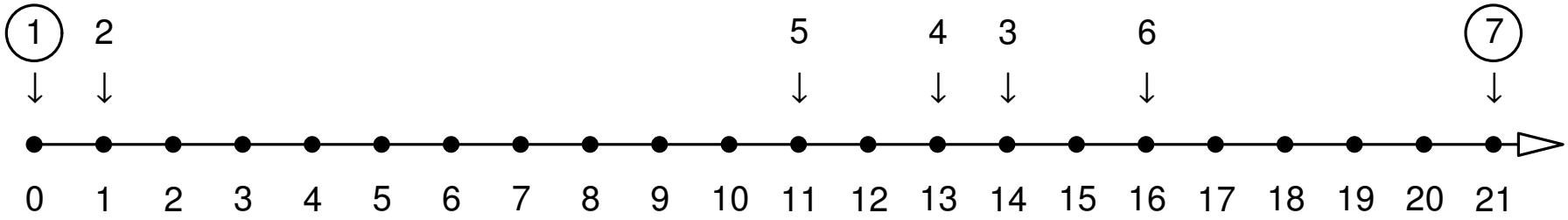
Par exemple, l'ordonnancement suivant est compatible :



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

Par exemple, l'ordonnancement suivant est compatible :

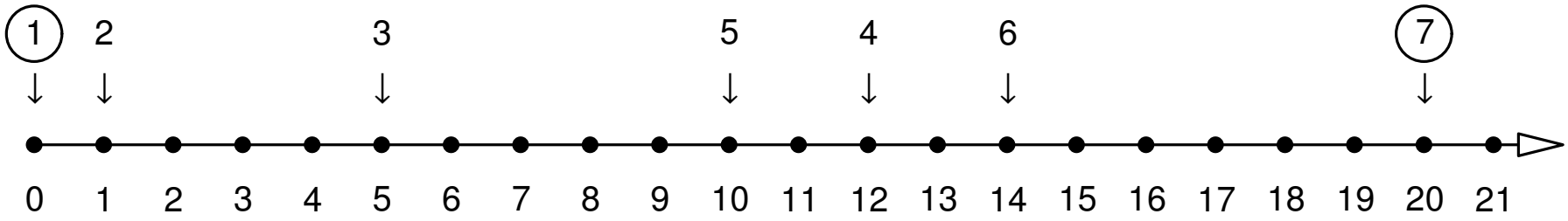


Cet ordonnancement s'écrit sous la forme compacte :

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.

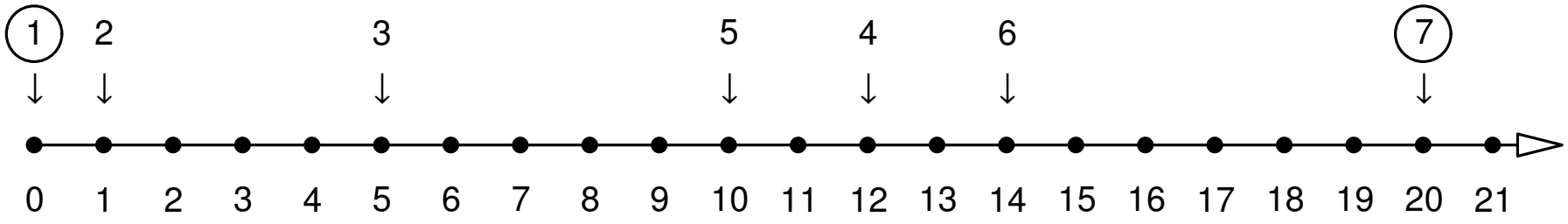
En revanche, l'ordonnancement suivant n'est **pas** compatible :



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

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

Il n'est pas compatible car on doit avoir plus de 5 jours entre le début de la tâche 5 et le début de la tâche 6.

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 :

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 δ après le début de la tâche A_i" :*

$$t_j - t_i \geq \delta.$$

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 δ après le début de la tâche A_i" :*

$$t_j - t_i \geq \delta.$$

- *contrainte au plus tard : "la tâche A_j commence au plus tard δ après le début de la tâche A_i" :*

$$t_j - t_i \leq \delta \quad \Leftrightarrow \quad t_i - t_j \geq -\delta.$$

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 δ après le début de la tâche A_i" :*

$$t_j - t_i \geq \delta.$$

- *contrainte au plus tard : "la tâche A_j commence au plus tard δ après le début de la tâche A_i" :*

$$t_j - t_i \leq \delta \quad \Leftrightarrow \quad t_i - t_j \geq -\delta.$$

- *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)" :*

$$t_i - t_1 \geq 0 \quad \text{et} \quad t_n - t_i \geq d_i.$$

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)	
(3) commence au plus tard 8 jours après le début de (4)	
(4) commence au plus tôt 1 jour après la fin de (2)	
(5) commence au plus tôt 1 jour avant la fin de (2)	
(6) commence au plus tôt après le début de (4)	
(6) commence au plus tôt après la fin de (5)	

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)	
(4) commence au plus tôt 1 jour après la fin de (2)	
(5) commence au plus tôt 1 jour avant la fin de (2)	
(6) commence au plus tôt après le début de (4)	
(6) commence au plus tôt après la fin de (5)	

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)	$t_4 - t_3 \geq -8$
(4) commence au plus tôt 1 jour après la fin de (2)	
(5) commence au plus tôt 1 jour avant la fin de (2)	
(6) commence au plus tôt après le début de (4)	
(6) commence au plus tôt après la fin de (5)	

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)	$t_4 - t_3 \geq -8$
(4) commence au plus tôt 1 jour après la fin de (2)	$t_4 - t_2 \geq 11 + 1 = 12$
(5) commence au plus tôt 1 jour avant la fin de (2)	
(6) commence au plus tôt après le début de (4)	
(6) commence au plus tôt après la fin de (5)	

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)	$t_4 - t_3 \geq -8$
(4) commence au plus tôt 1 jour après la fin de (2)	$t_4 - t_2 \geq 11 + 1 = 12$
(5) commence au plus tôt 1 jour avant la fin de (2)	$t_5 - t_2 \geq 11 - 1 = 10$
(6) commence au plus tôt après le début de (4)	
(6) commence au plus tôt après la fin de (5)	

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)	$t_4 - t_3 \geq -8$
(4) commence au plus tôt 1 jour après la fin de (2)	$t_4 - t_2 \geq 11 + 1 = 12$
(5) commence au plus tôt 1 jour avant la fin de (2)	$t_5 - t_2 \geq 11 - 1 = 10$
(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)	

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)	$t_4 - t_3 \geq -8$
(4) commence au plus tôt 1 jour après la fin de (2)	$t_4 - t_2 \geq 11 + 1 = 12$
(5) commence au plus tôt 1 jour avant la fin de (2)	$t_5 - t_2 \geq 11 - 1 = 10$
(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)	$t_6 - t_5 \geq 0 + 5 = 5$

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 $f(x_i, x_j) = d$.*

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.*

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

Définition (Graphe potentiel-tâches)

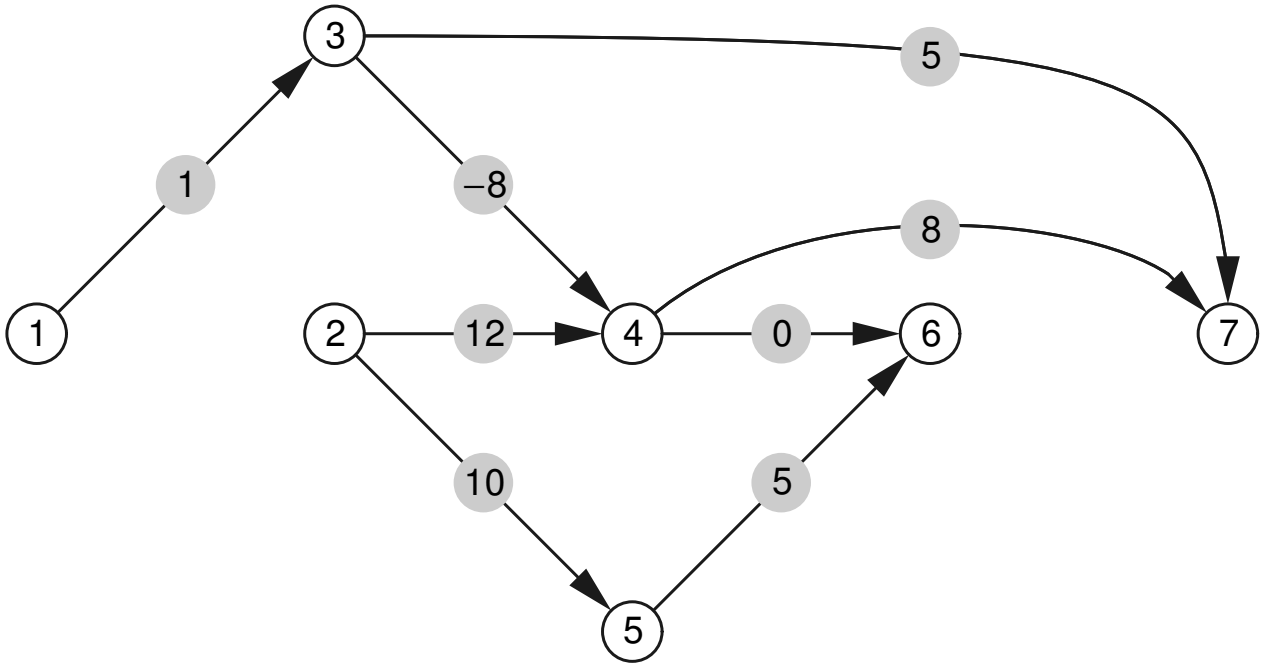
On associe à un problème d'ordonnancement un graphe orienté et valué G = (S,A, f) tel que :

- *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 $f(x_i, x_j) = d$.*

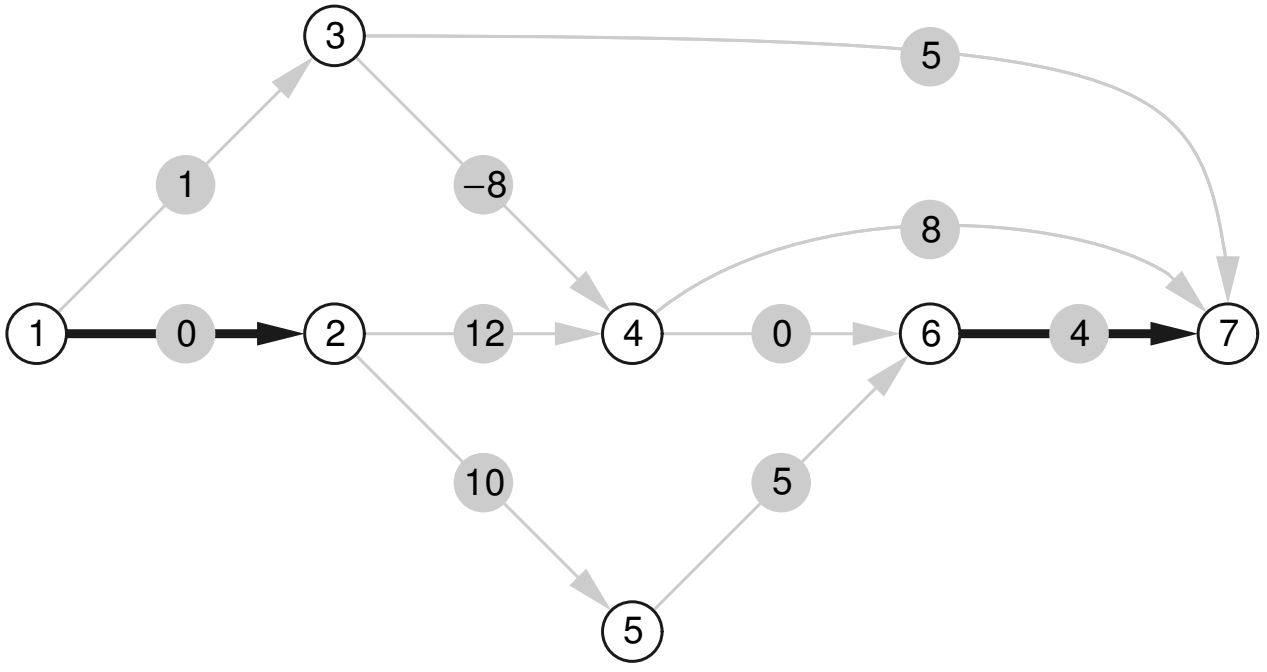
On ajoute éventuellement des **contraintes implicites** *pour que chaque tâche dans le graphe appartienne à au moins un chemin reliant la tâche de début de projet et celle de fin de projet.*

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.

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



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



Il faut ajouter deux arcs correspondant à des contraintes implicites :

- A_2 ne peut commencer qu'après le début du projet

$$t_2 - t_1 \geq 0$$

- A_7 commence au plus tôt à la fin de A_6

$$t_7 - t_6 \geq d_6 = 4$$

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.*

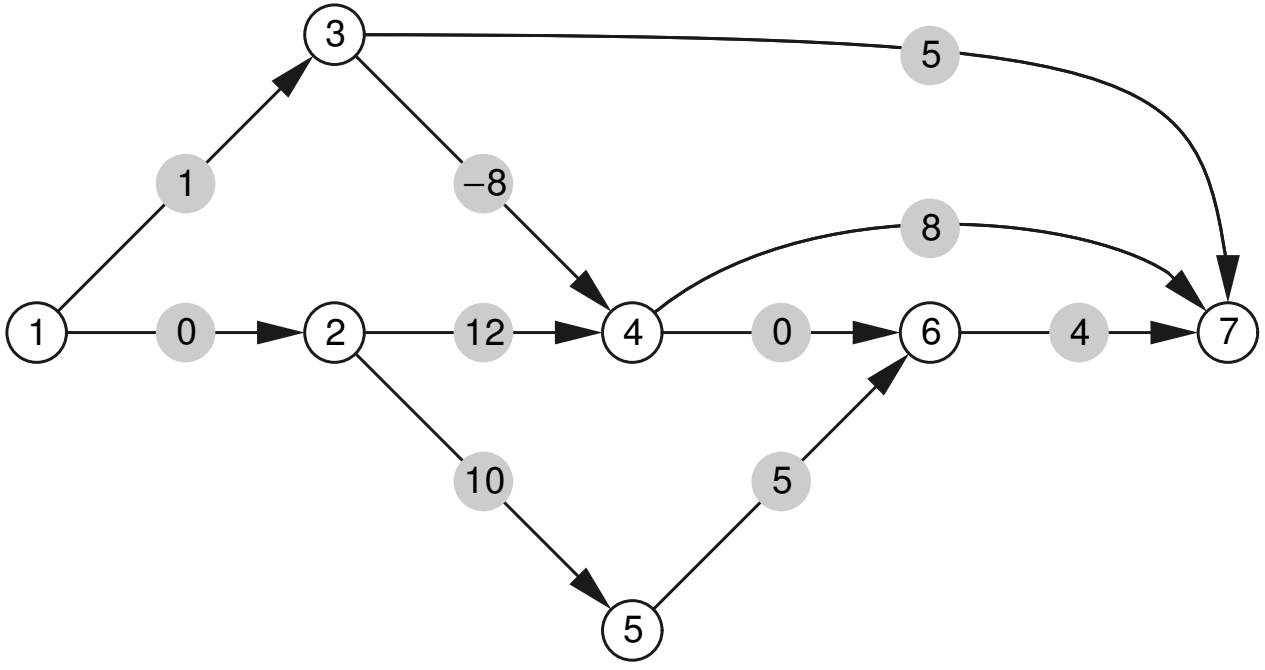
Pour déterminer l'ordonnancement au plus tôt d'un projet, il suffit de déterminer **les chemins les plus longs dans le graphe** *G depuis le sommet correspondant au début du projet. Les distances* $\{d_i^{(-)}\}_{i=1,\dots,n}$ *ainsi obtenues sont les dates de commencement au plus tôt de chaque tâche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

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

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. □

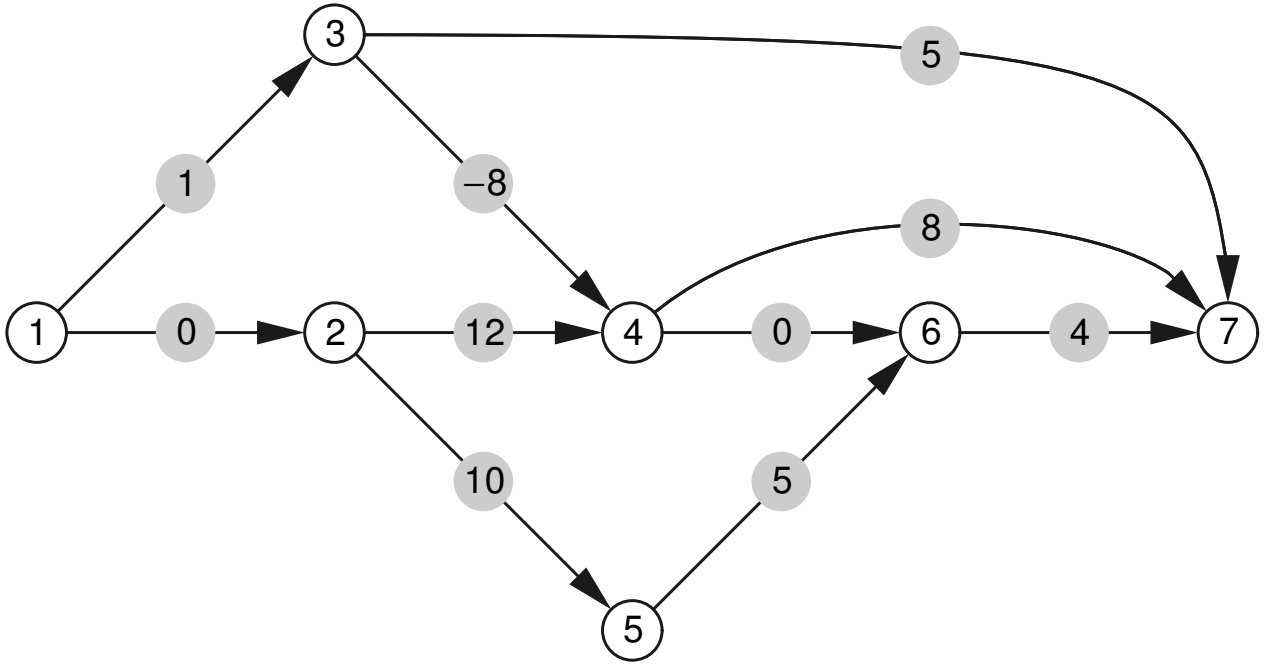
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ ($P(i)$)							

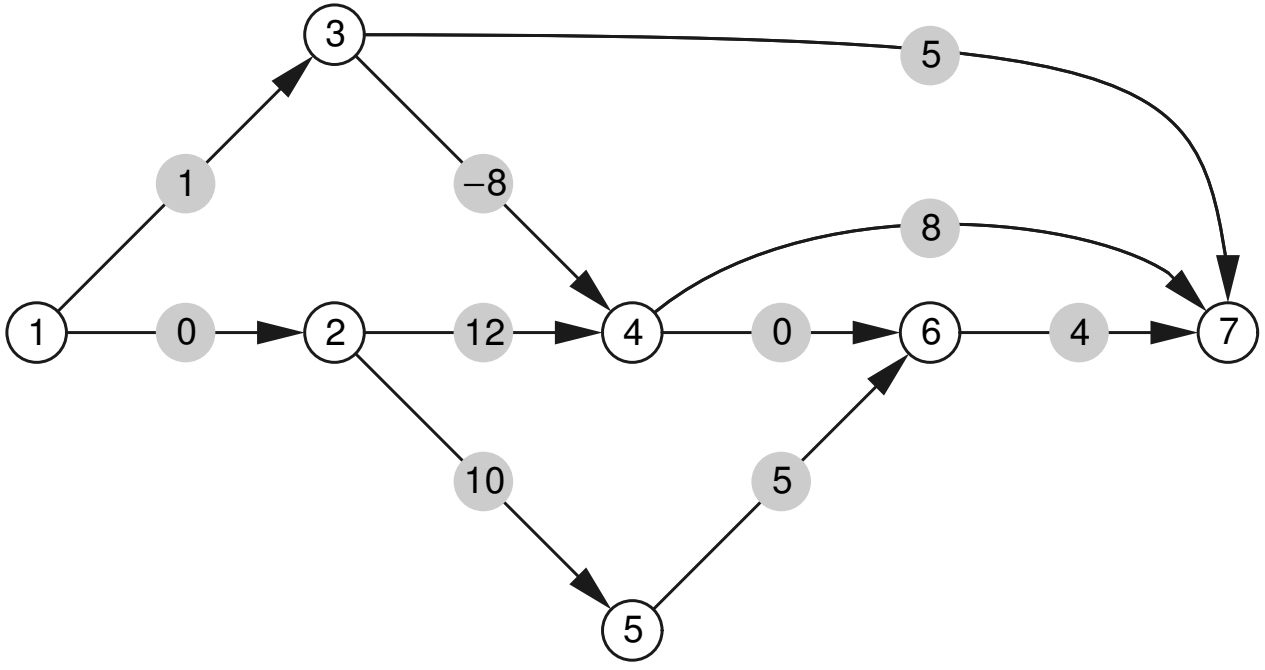
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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(∅)						

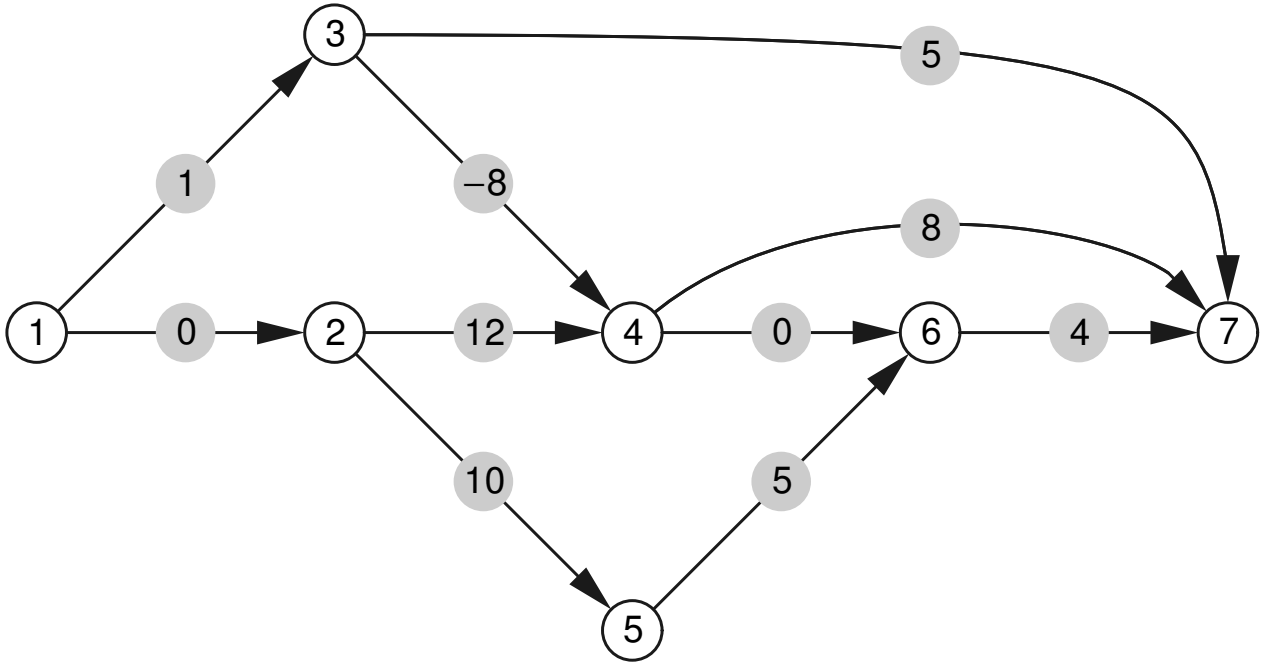
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

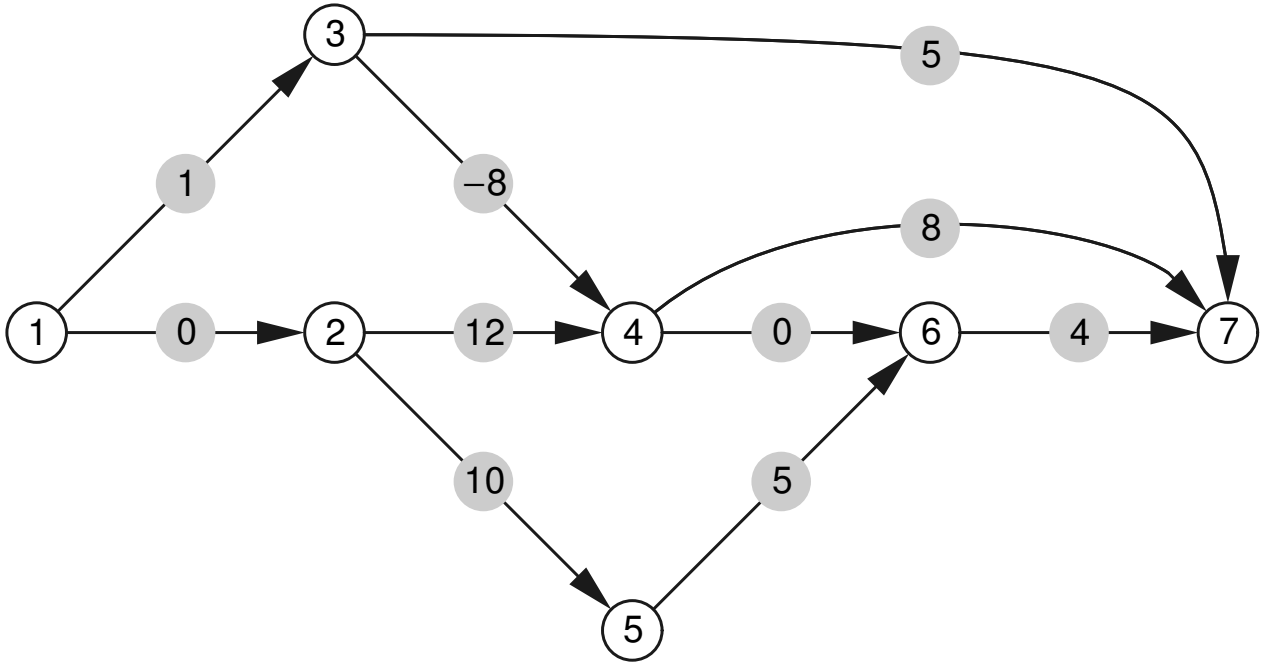
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

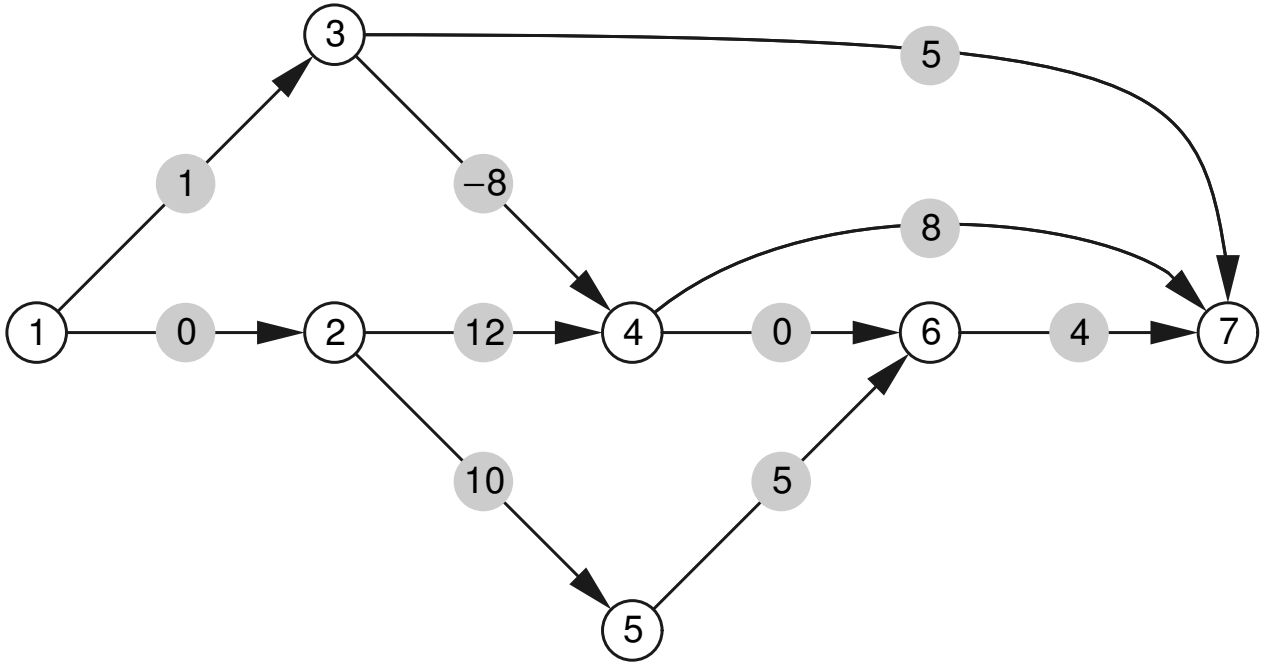
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

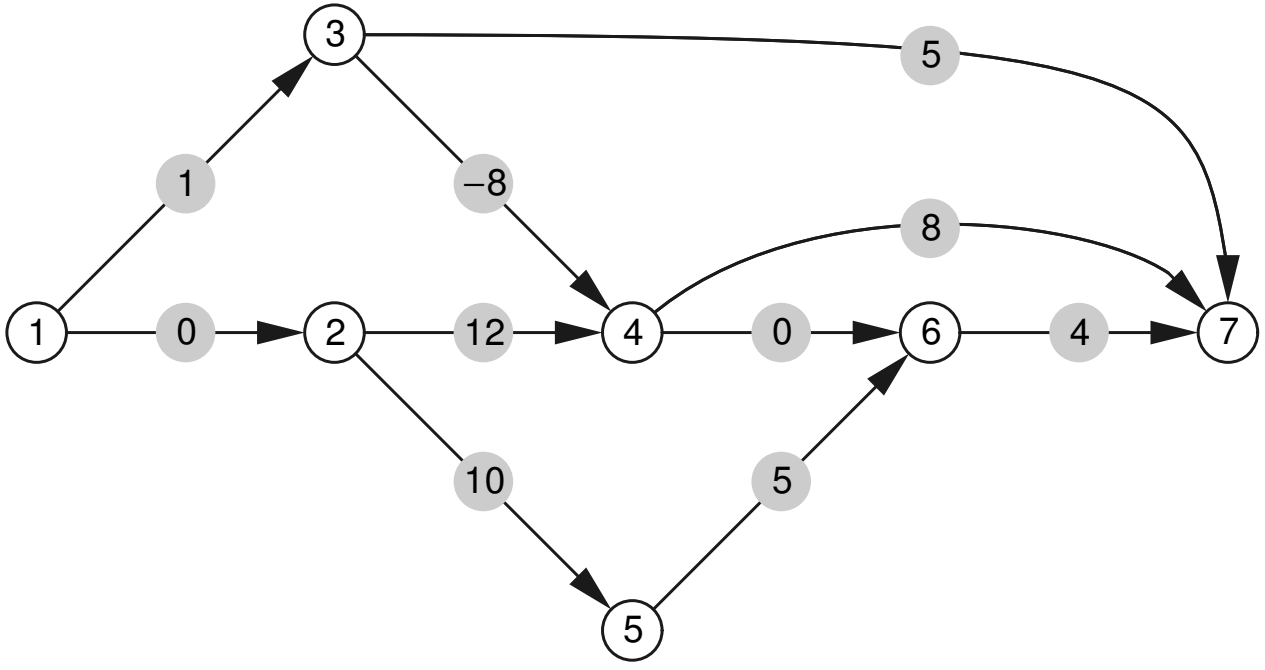
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$1(1)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

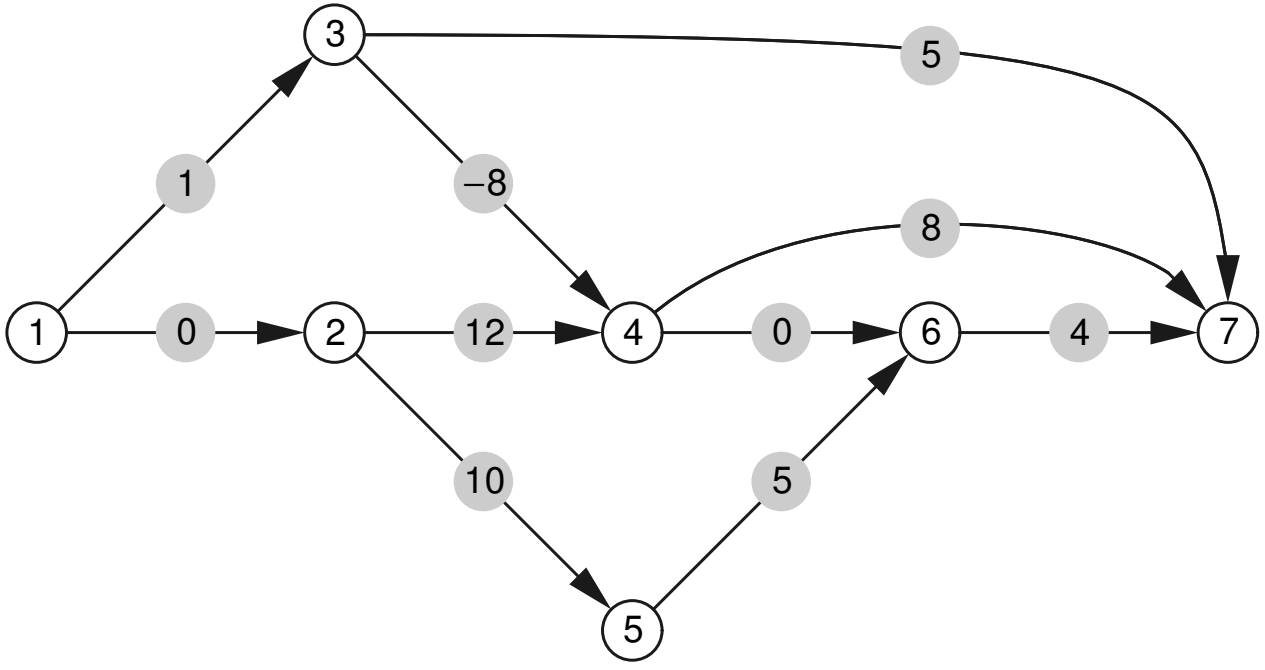
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$1(1)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

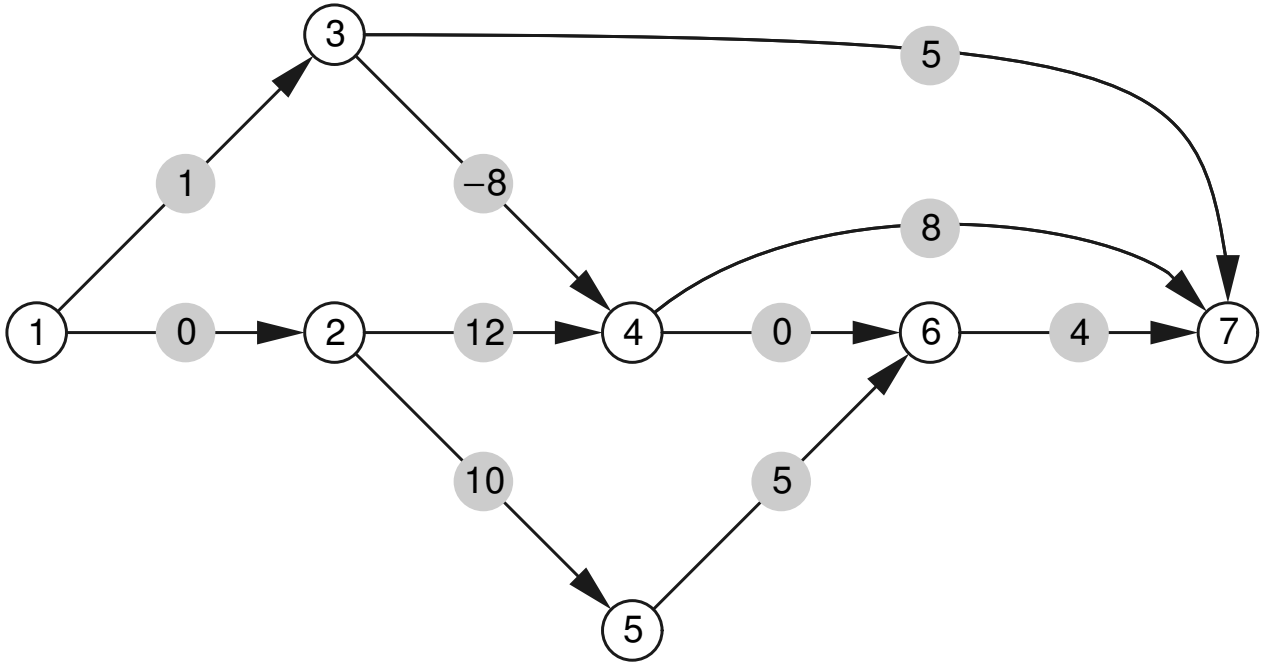
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$1(1)$	$12(2)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

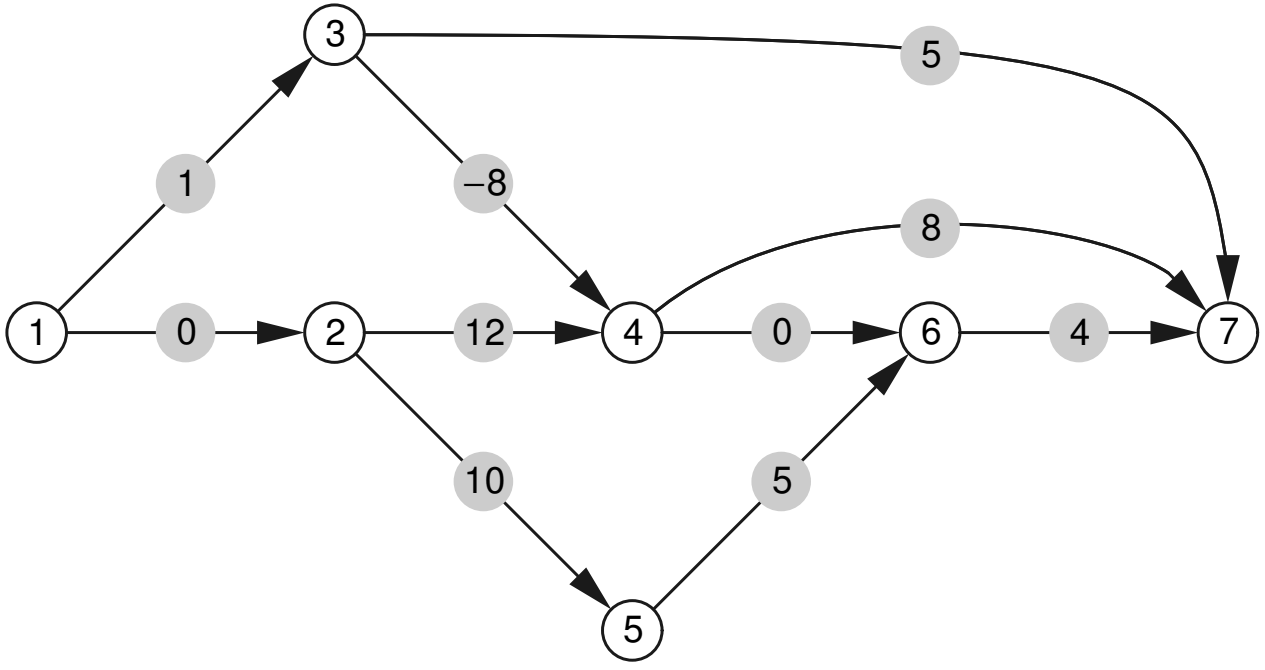
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$1(1)$	$12(2)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

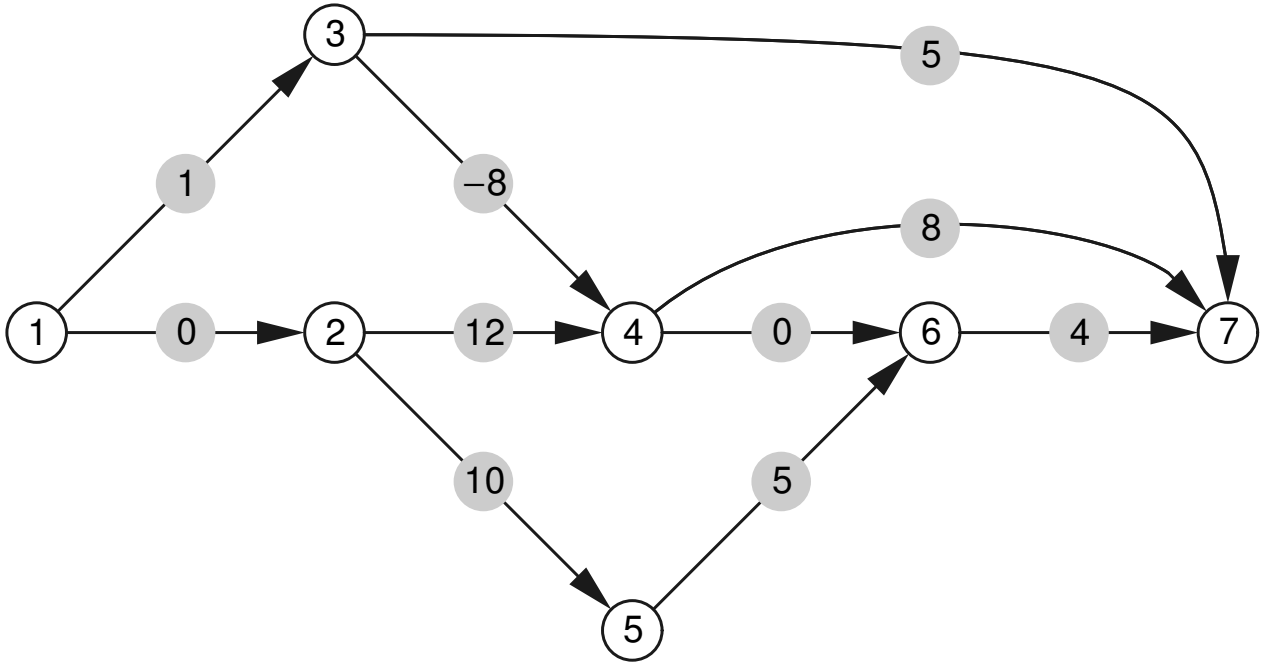
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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
$d(i)$ $(P(i))$	$0(\emptyset)$	$0(1)$	$1(1)$	$12(2)$	$10(2)$	$-\infty(\emptyset)$	$-\infty(\emptyset)$

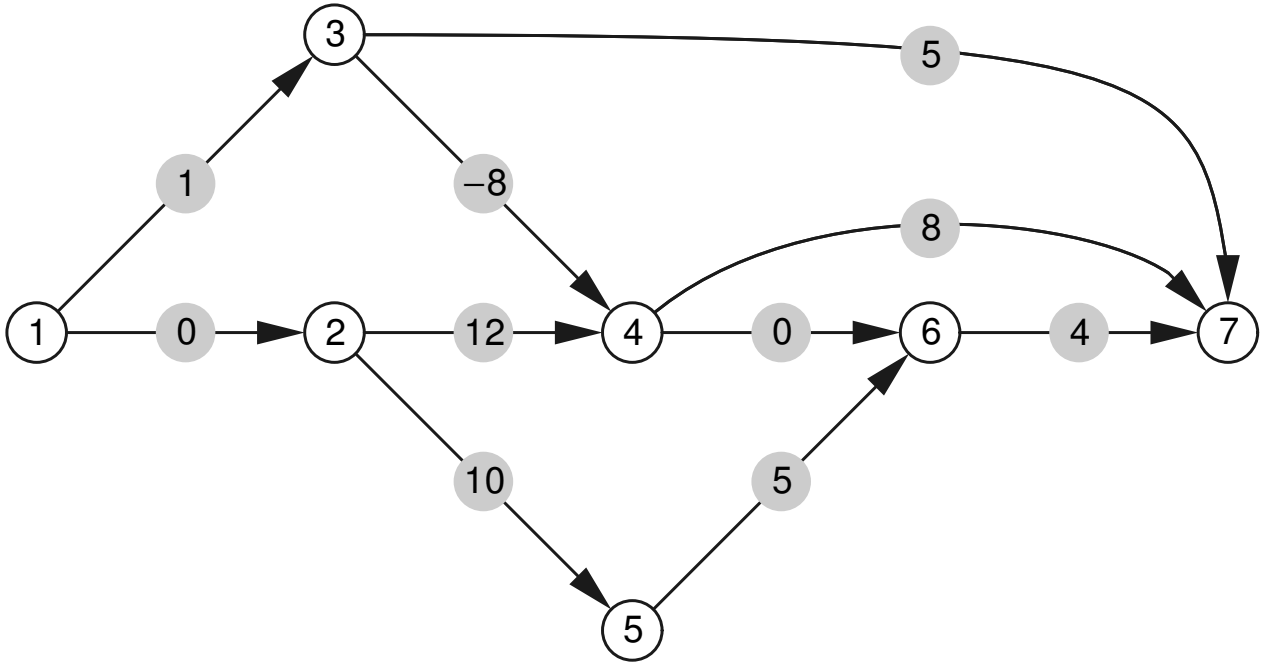
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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(1)	1(1)	12(2)	10(2)	$-\infty(\emptyset)$	$-\infty(\emptyset)$

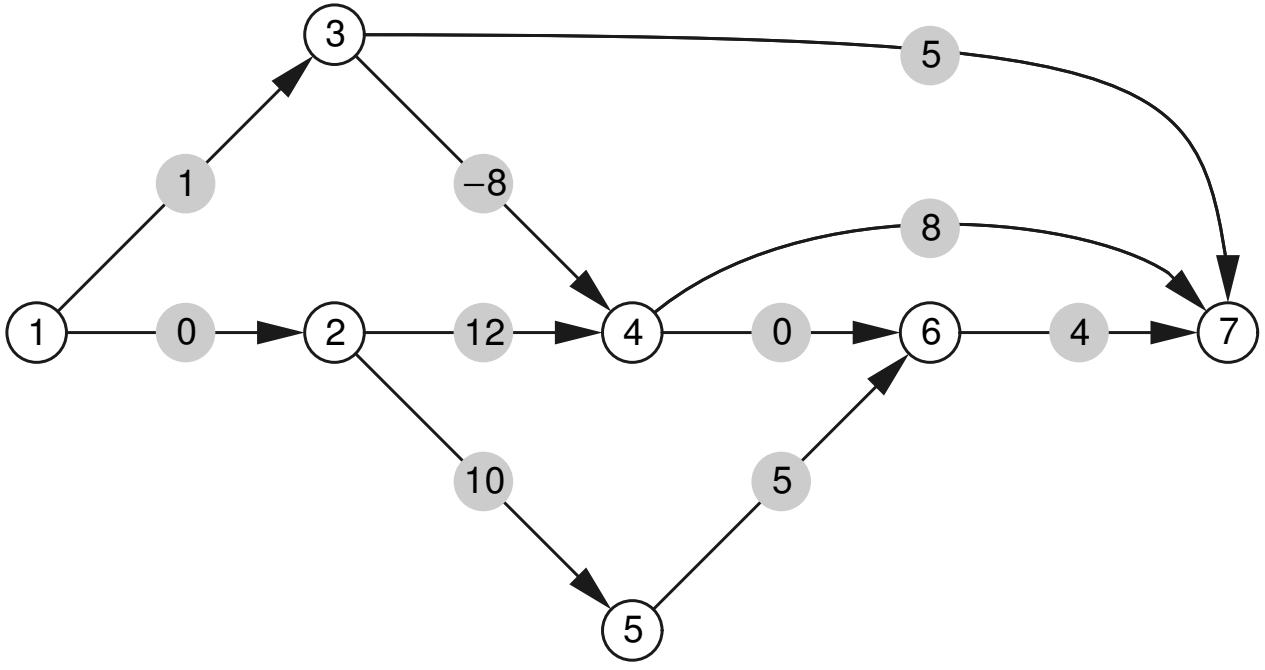
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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(1)	1(1)	12(2)	10(2)	15(5)	$-\infty$(∅)

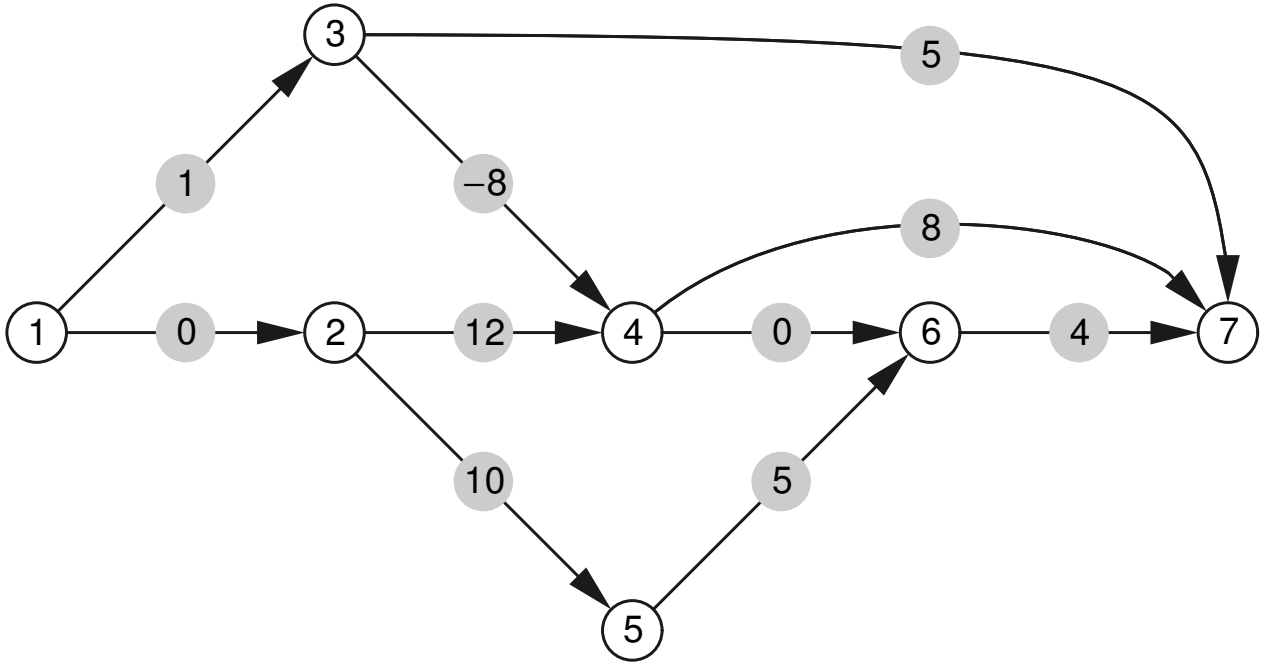
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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(1)	1(1)	12(2)	10(2)	15(5)	$-\infty$(∅)

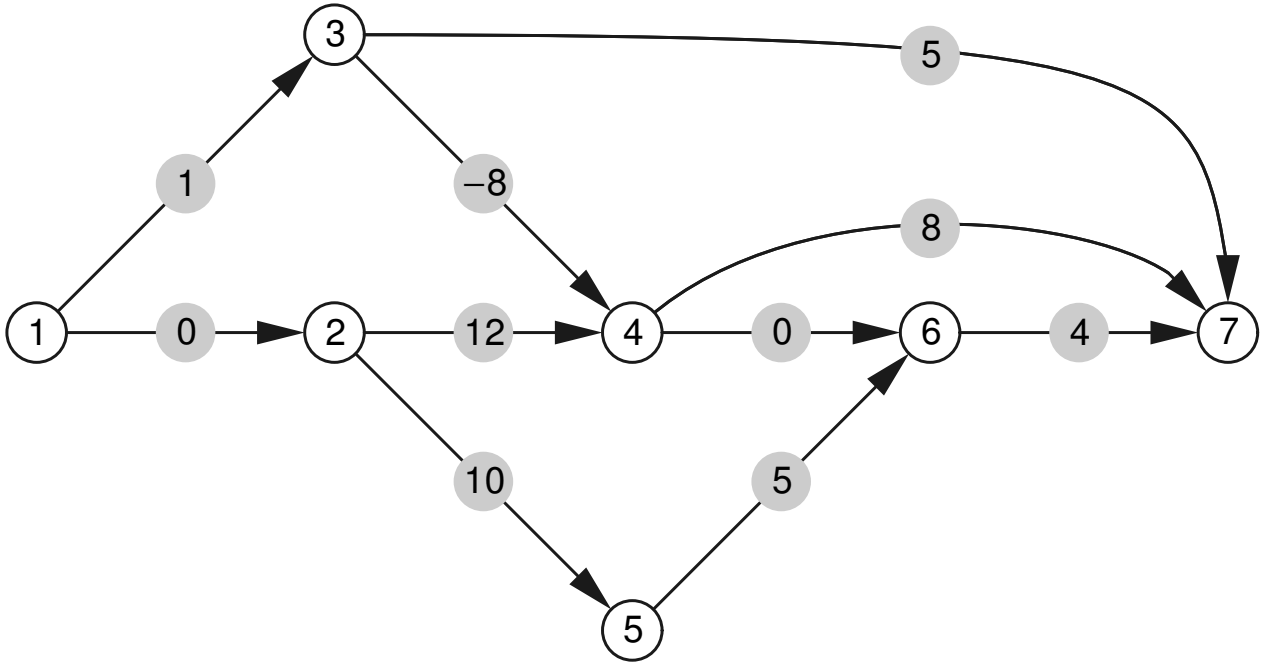
Déterminons l'ordonnancement au plus tôt du projet. Le graphe associé est :



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(1)	1(1)	12(2)	10(2)	15(5)	20(4)

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(1)	1(1)	12(2)	10(2)	15(5)	20(4)

Les dates de début de tâche $t_i^{(-)}$ 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
$t_i^{(-)} = d(i)$	0	0	1	12	10	15	20

La durée minimale de réalisation du projet est donc de

$$T_{\text{min}} =$$

Les dates de début de tâche $t_i^{(-)}$ 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
$t_i^{(-)} = d(i)$	0	0	1	12	10	15	20

La durée minimale de réalisation du projet est donc de

$$T_{\min} = t_7^{(-)} = 20 \text{ jours.}$$