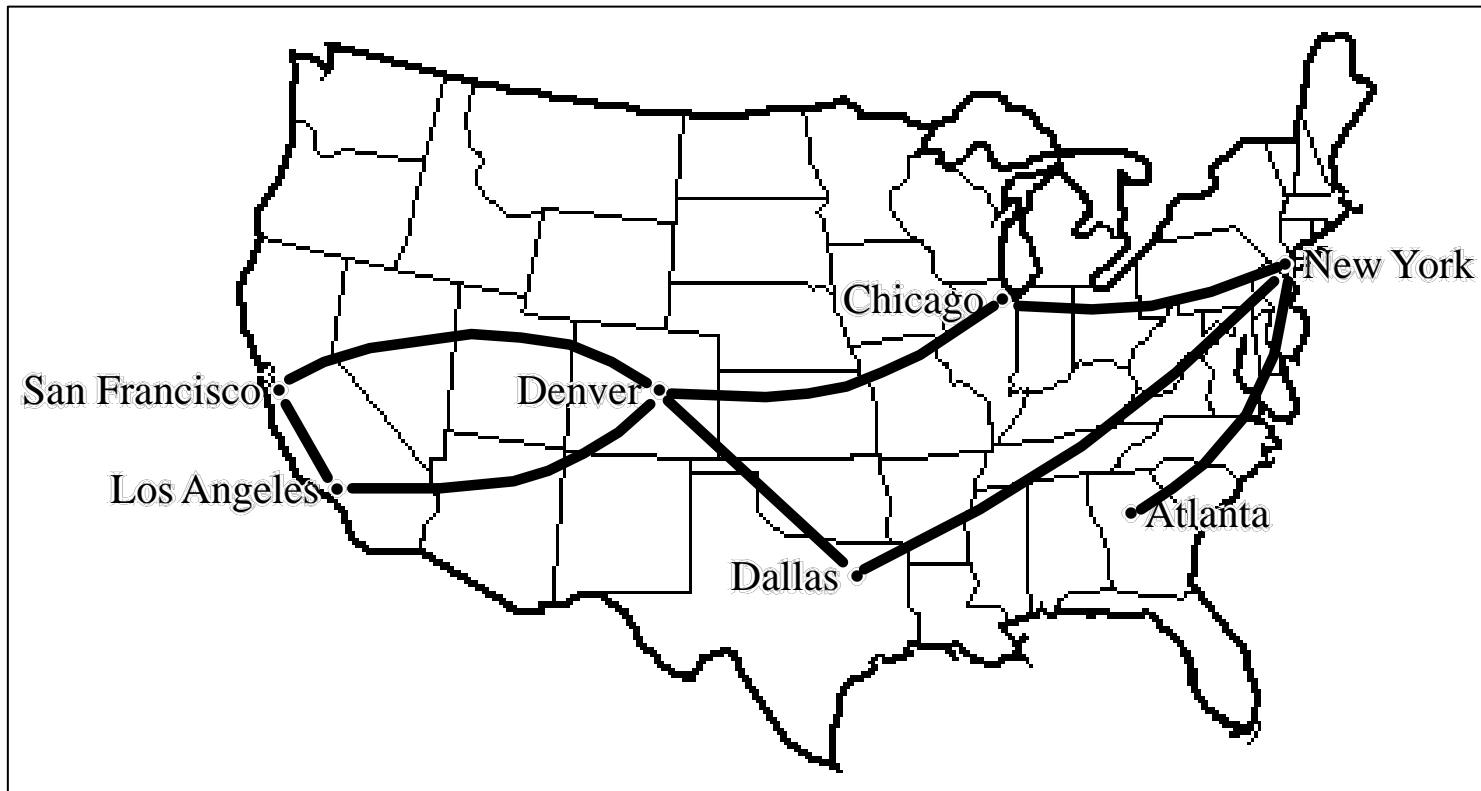


Plan du cours

1. Introduction
 - Contenu du cours
2. Logique mathématique
 - Calcul propositionnel
 - Calcul des prédicats
 - Logique floue et aide à la décision
3. Induction
4. Analyse d'algorithmes
 - Comparaison asymptotique de fonctions
 - Complexité
5. Récurrence
6. Mathématique de la gestion
 - Théorie des graphes
 - Optimisation

1. Graphe ?

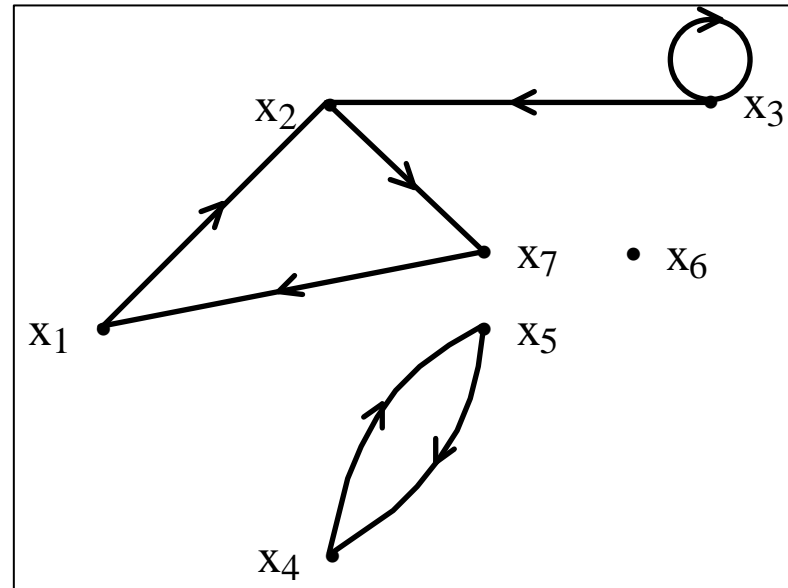


Graphe orienté

$$G = (X, U)$$

- X : ensemble fini d'éléments appelés **sommets**,
- U : sous-ensemble de $X \times X$ (couples de sommets) dont les éléments sont appelés **arcs**.

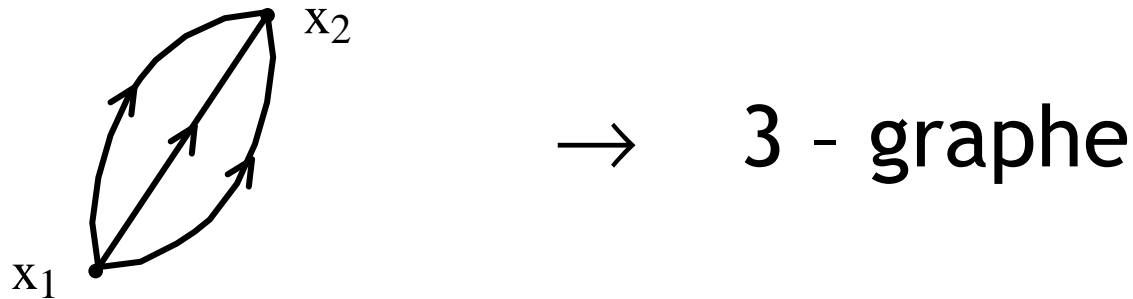
Exemple 1



- $X = \{ x_1 , x_2 , x_3 , x_4 , x_5 , x_6 , x_7 \}$
- $U = \{ (x_1, x_2) , (x_2, x_7) , (x_3, x_3) , (x_3, x_2) , (x_4, x_5) , (x_5, x_4) , (x_7, x_1) \}$

P-graphes

- U = famille de couples de sommets de $X \times X$, avec au plus p duplications.



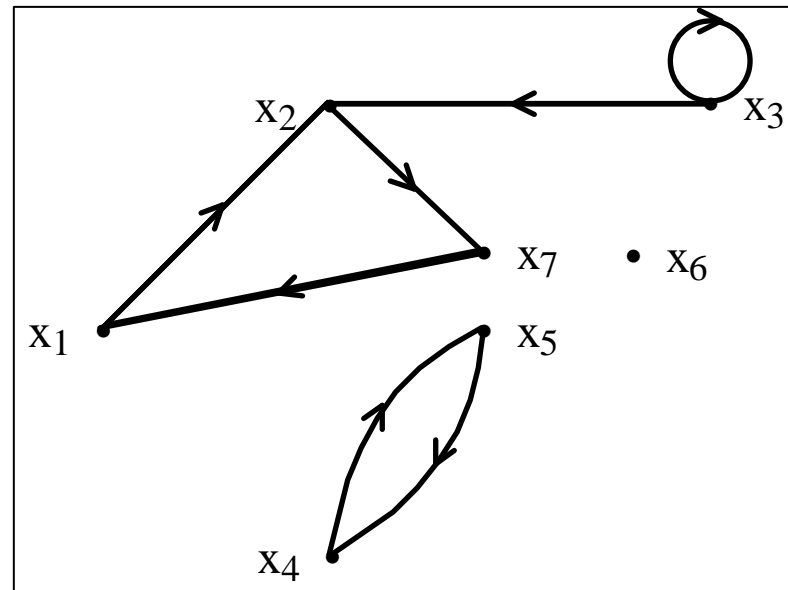
- Dans le cours : 1-graphes uniquement !

Terminologie

- **Chaîne** : Suite de sommets telle que si x_k et x_l sont deux sommets consécutifs de cette suite, alors (x_k, x_l) ou $(x_l, x_k) \in U$.
- **Chemin** : Suite de sommets telle que si x_k et x_l sont deux sommets consécutifs de cette suite, alors $(x_k, x_l) \in U$.
- **Cycle** : Chaîne dont le dernier sommet coïncide avec le premier.
- **Circuit** : Chemin dont le dernier sommet coïncide avec le premier.

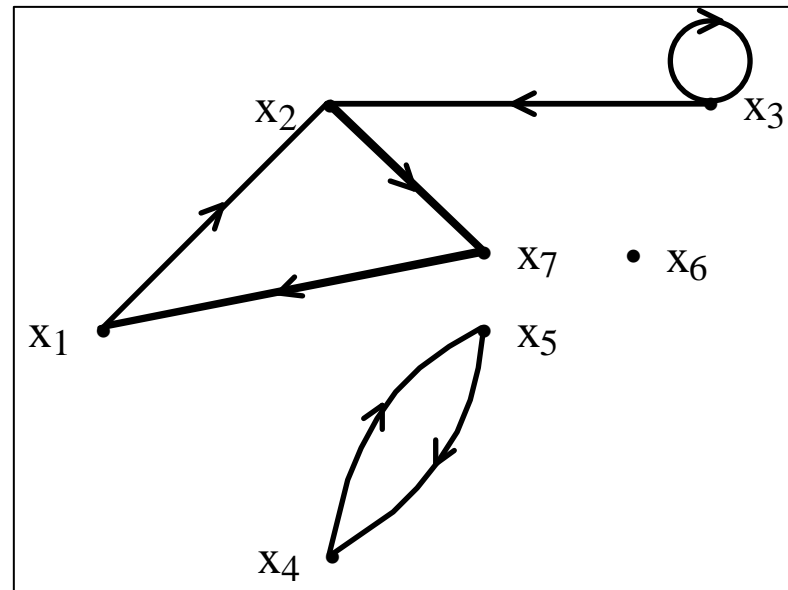
Exemple 1

Chaîne

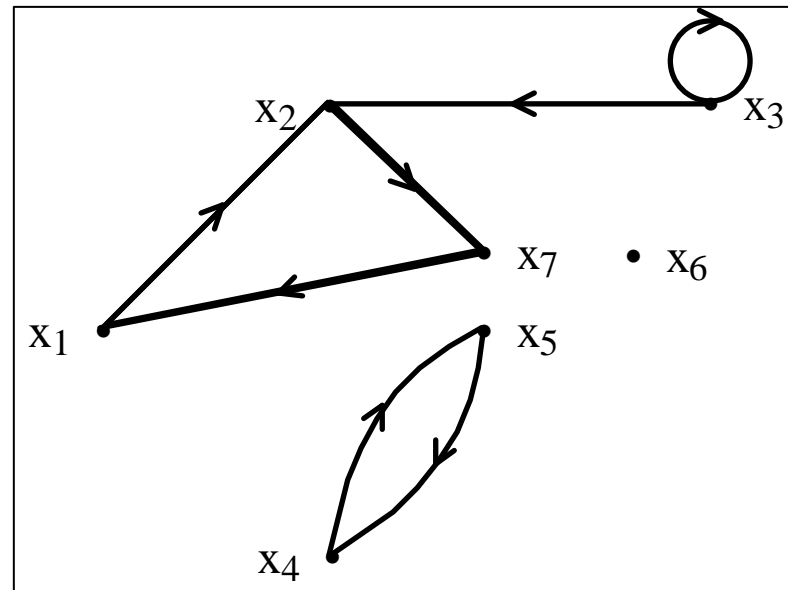


Exemple 1

Chemin



Exemple 1



Circuit

Terminologie (2)

- **Circuits particuliers :**
 - Circuit élémentaire,
 - Circuit hamiltonien (1! par chaque sommet),
 - Circuit eulérien (1! par chaque arc).
- **Graphe connexe :**
 $\forall x, y \in X$ avec $x \neq y \exists$ chaîne entre x et y .
- **Graphe fortement connexe :**
 $\forall x, y \in X$ avec $x \neq y \exists$ chemin de x vers y .

Terminologie (3)

- **Sous-graphe** : engendré par $Y \subset X$:
graphe (Y, U_Y) où $U_Y \subset U$ représente les arcs de U qui ont leurs extrémités dans Y .
- **Graphe partiel** : engendré par $V \subset U$:
graphe (X, V) .
- **Matrice d'adjacence** : $\mathbf{A} = (a_{ij}) \quad n \times n$

$$\begin{cases} a_{ij} = 1 & \text{ssi } (x_i, x_j) \in U \\ a_{ij} = 0 & \text{ssi } (x_i, x_j) \notin U \end{cases}$$

Example 1

$$\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Terminologie (4)

- Ensemble des **successeurs** de $x \in X$:

$$\Gamma^+(x) = \{y \in X \mid (x, y) \in U\}$$

- Ensemble des **prédécesseurs** de $x \in X$:

$$\Gamma^-(x) = \{y \in X \mid (y, x) \in U\}$$

- **Graphe valué** : à chaque arc est associée une valeur (nombre).

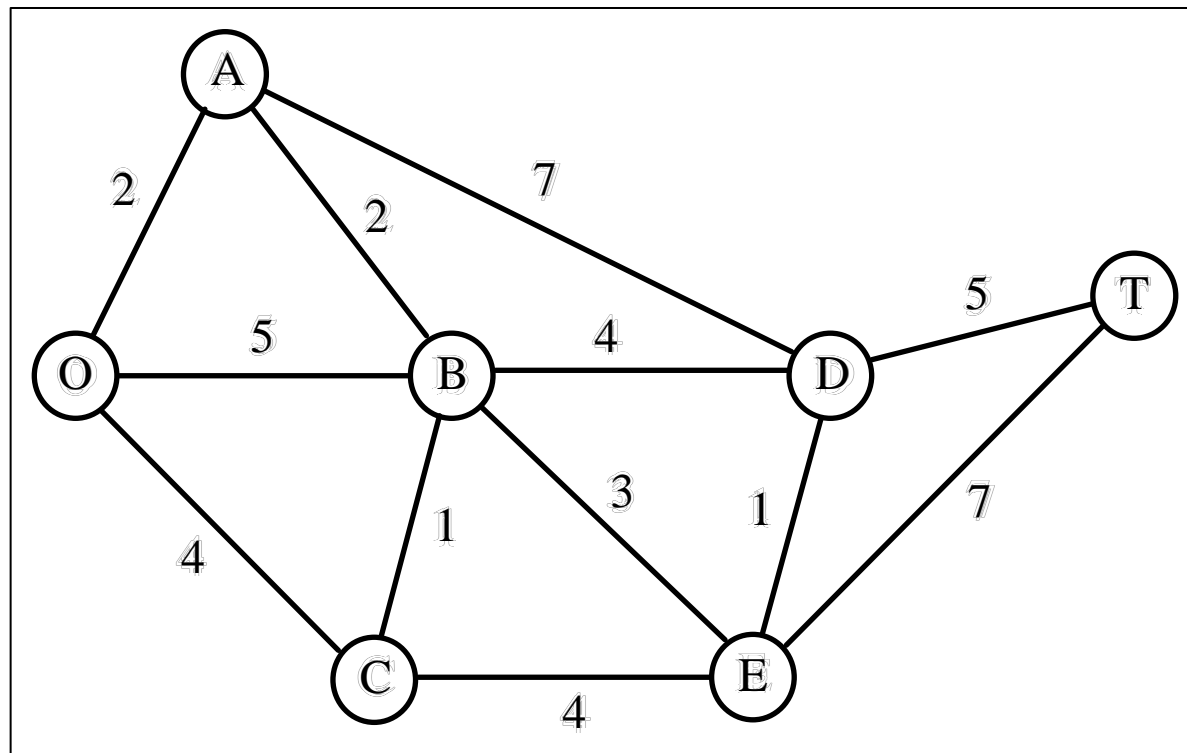
Graphe non orienté

$$G = (X, E)$$

- X : ensemble fini d'éléments appelés **sommets**,
- E : ensemble de paires de sommets dont les éléments sont appelés **arêtes**.
- **Graphe simple** :
 - Au plus une arête entre deux sommets,
 - Pas de boucles.
- **Graphe orienté symétrique**.

Exemple 2

- Dans une réserve naturelle :
 - Sommets : Postes de surveillance/attractions (O: entrée).
 - Arêtes : Routes.
 - Valeurs : Longueurs des routes (km).



2. Niveaux, rangs et circuits

- Dans un graphe orienté.
- Niveaux :

$$\left\{ \begin{array}{l} X(0) = \{x \in X : \Gamma^+(x) = \emptyset\} \\ X(1) = \{x \in X \setminus X(0) : \Gamma^+(x) \subseteq X(0)\} \\ X(2) = \{x \in X \setminus (X(0) \cup X(1)) : \Gamma^+(x) \subseteq X(0) \cup X(1)\} \\ \vdots \\ X(k) = \{x \in X \setminus (X(0) \cup \dots \cup X(k-1)) : \Gamma^+(x) \subseteq X(0) \cup \dots \cup X(k-1)\} \\ \vdots \end{array} \right.$$

Propriétés des niveaux

- Disjoints 2 à 2.
- Réunion = X ssi graphe sans circuits.
- $x \in X(k)$ ssi le plus long chemin issu de x est de longueur k (longueur = nombre d'arcs).

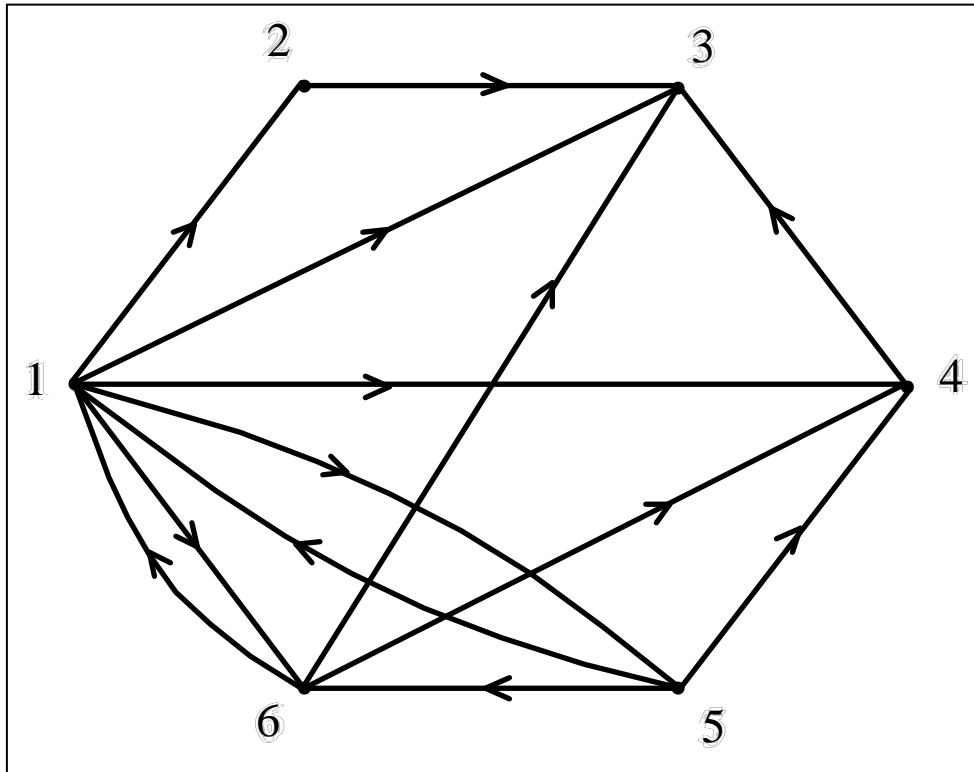
Algorithme pour l'obtention des niveaux

1. Ecrire la matrice d'adjacence et poser $k = 0$.
2. $X(k) = \{ \text{sommets correspondants aux lignes non barrées ne contenant que des 0 ou des 1 barrés} \}$
 - Si $X(k) = \emptyset$, aller en 4.
 - Sinon, barrer les lignes et colonnes correspondant à $X(k)$.
3. $k := k + 1$ et retourner en 2.
4. $X(0), X(1), \dots, X(k-1)$ sont les k niveaux du graphe.

Détection des circuits

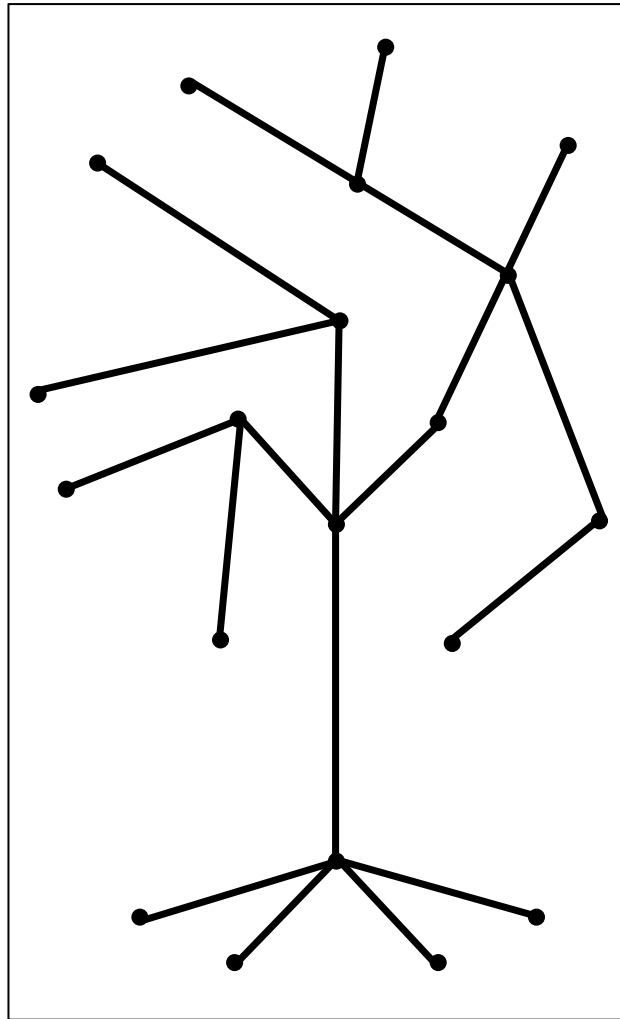
- Si toutes les lignes ne sont pas barrées à la fin de l'algorithme, il existe au moins un circuit passant par les sommets non barrés.
- **Rangs** : Comme pour les niveaux mais en considérant les prédécesseurs.

Exemple 3



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1	1	1	1	1
x_2	0	0	1	0	0	0
x_3	0	0	0	0	0	0
x_4	0	0	1	0	0	0
x_5	1	0	0	1	0	1
x_6	1	0	1	1	0	0

3. L'arbre partiel minimum



Arbre

- Définition : Un arbre est un graphe simple sans cycle et tel qu'il existe une chaîne entre toute paire de sommets.
- Propriétés :
 1. \exists au moins un sommet se trouvant sur une seule arête.
 2. n sommets $\leftrightarrow n-1$ arêtes.
 3. Addition d'une arête \leftrightarrow création d'un et un seul cycle.
 4. Chaque paire de sommets est reliée par un et une seule chaîne. Devient faux si l'on enlève une arête.

Arbre partiel minimum

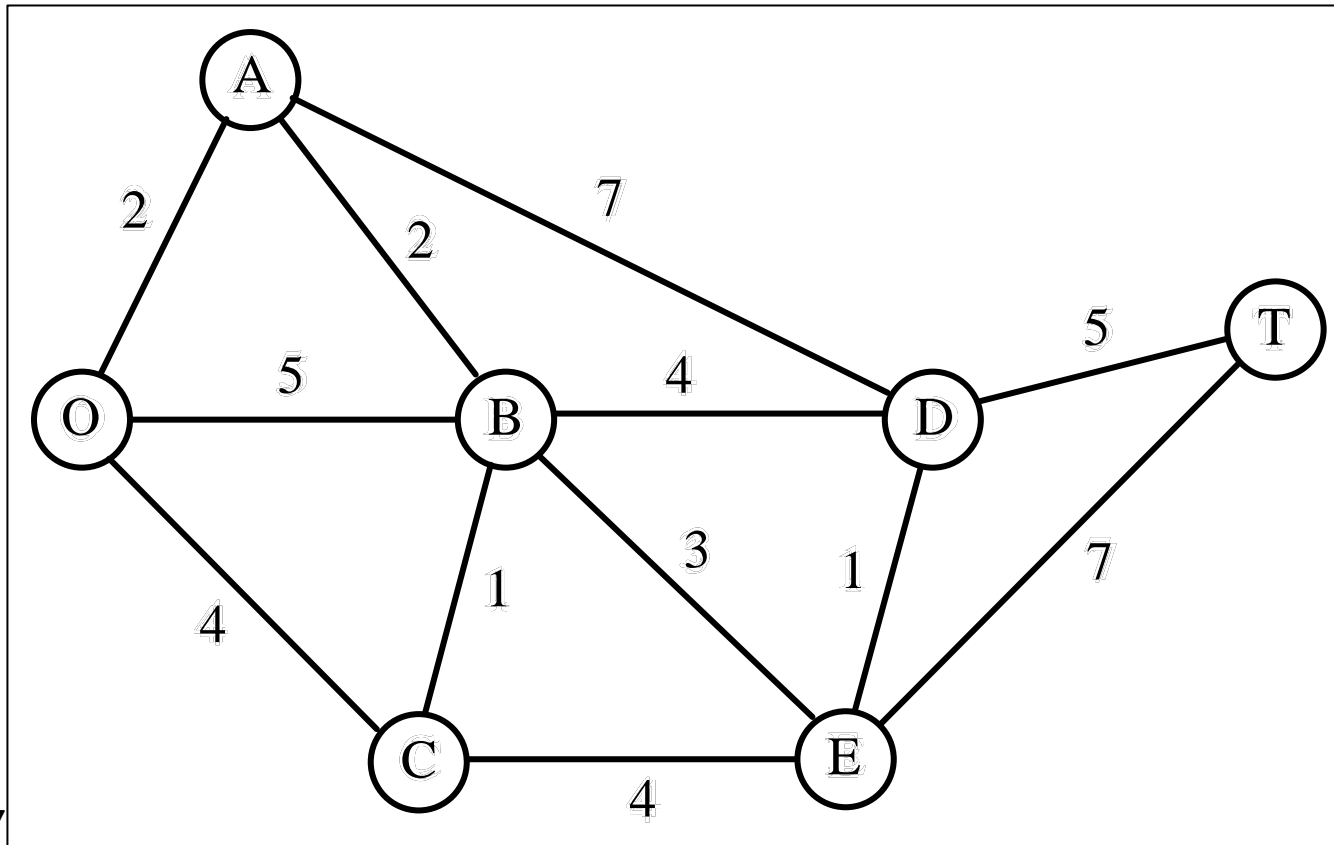
- Dans un graphe simple valué.
- Graphe partiel tel que :
 - Arbre,
 - Somme des valeurs des arêtes est minimum.
- Théorème de base :
 - Suppositions : graphe complet, arêtes de valeurs toutes différentes.
 - Énoncé : $\forall A \subset X$, soit $E_A = \{ \text{arêtes ayant une extrémité dans } A \text{ et l'autre dans } X \setminus A \}$: l'arête de E_A de valeur minimum appartient à l'arbre partiel minimum.

Algorithme de Prim

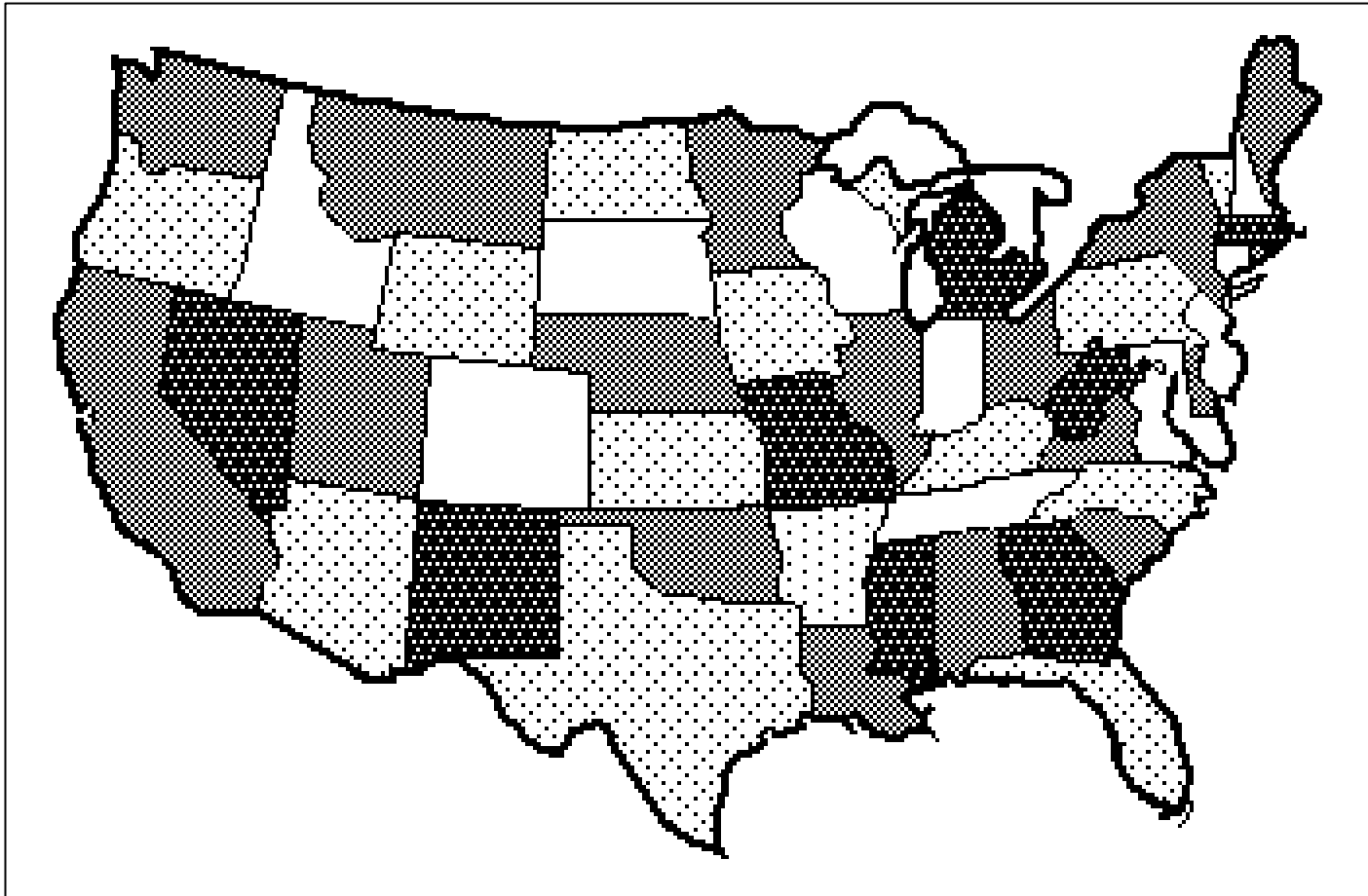
1. Choisir un sommet arbitrairement et marquer l'arête de valeur minimum issue de ce sommet.
2. Soit $A = \{ \text{sommets extrémités des arêtes marquées} \}$
Si $A = X$, aller en 4.
3. Marquer l'arête de valeur minimum joignant un sommet de A à un sommet de $X \setminus A$ et aller en 2.
4. L'ensemble des arêtes marquées constitue un arbre partiel minimum.

Exemple 2

- Relier les 7 postes du parc en réseau.
- Minimiser la longueur du câblage.



4. Problèmes de coloration



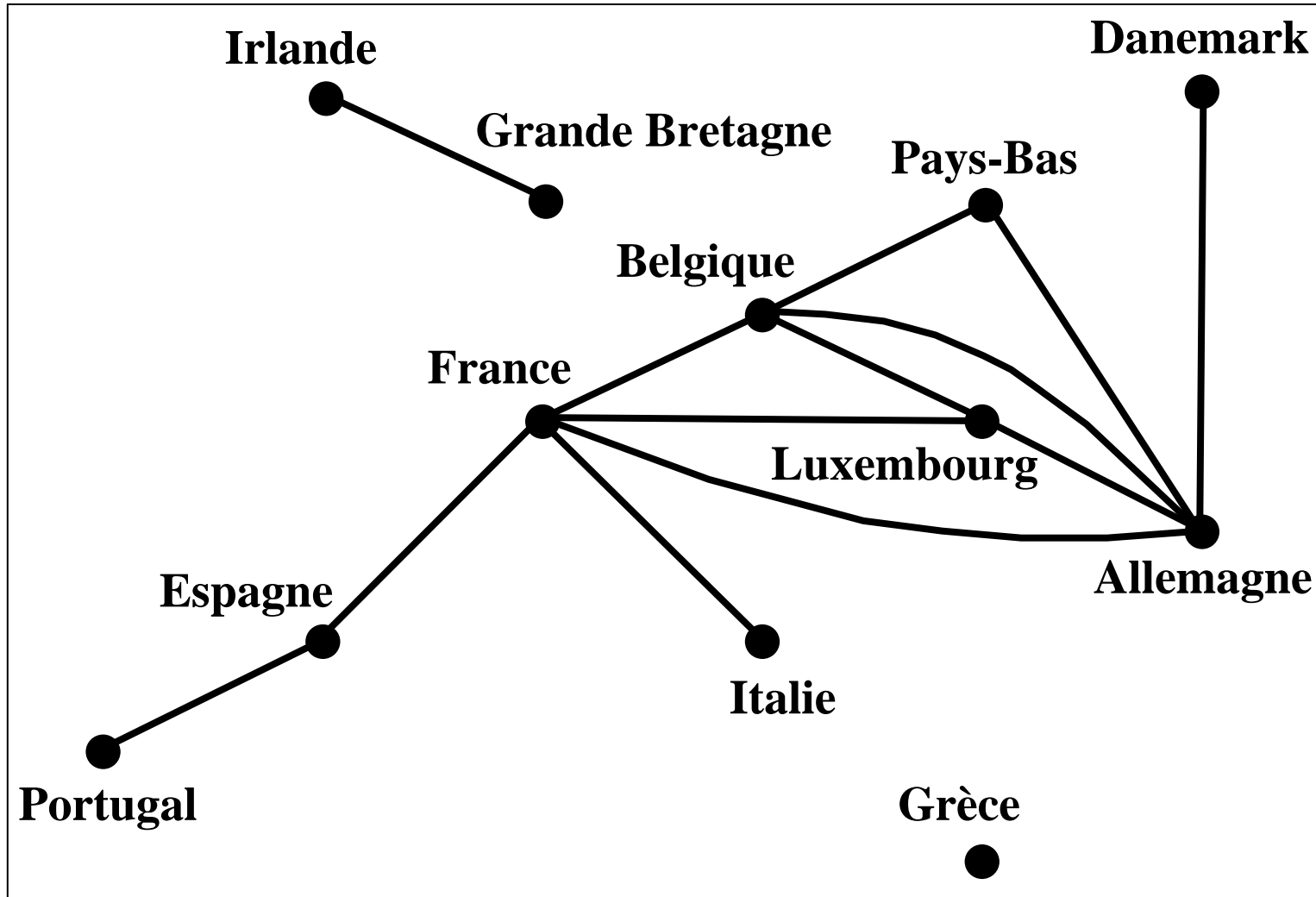
Coloration des sommets d'un graphe

- Attribution d'une couleur à chaque sommet de telle sorte que deux sommets adjacents aient des couleurs différentes.
- **Nombre chromatique** : Nombre minimum de couleurs nécessaires pour colorer un graphe.
 - Obtention : problème difficile → heuristiques.

Algorithme de Welsh et Powell

1. Numérotter les sommets par ordre décroissant de leur degré
(degré = nombre d'arêtes issues d'un sommet).
Poser $i = 1$ (couleur) et $N = X$ (sommets non encore colorés).
2. Donner la couleur i au sommet de N qui a le plus petit numéro.
3. Soit $N_i = \{ \text{sommets non colorés non adjacents à un sommet de couleur } i \}$.
Si $N_i \neq \emptyset$, poser $N = N_i$ et aller en 2.
4. Poser $N = \{ \text{sommets non encore colorés} \}$.
Si $N \neq \emptyset$, poser $i := i+1$ (couleur suivante) et aller en 2.
Sinon, on a une coloration en i couleurs.

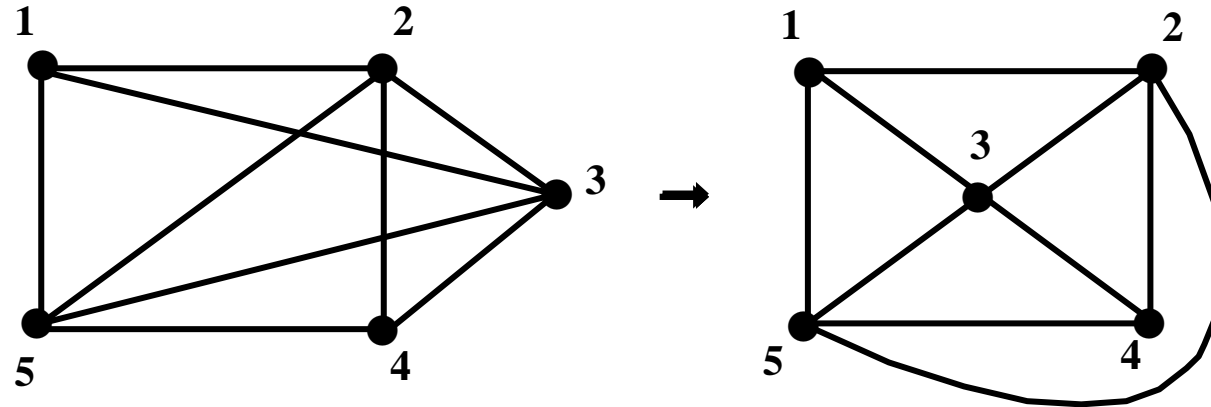
Exemple 4



Applications

- Coloration de cartes.
- Problèmes d'horaires.
- Chargement de camions.
- Allocation de ressources.

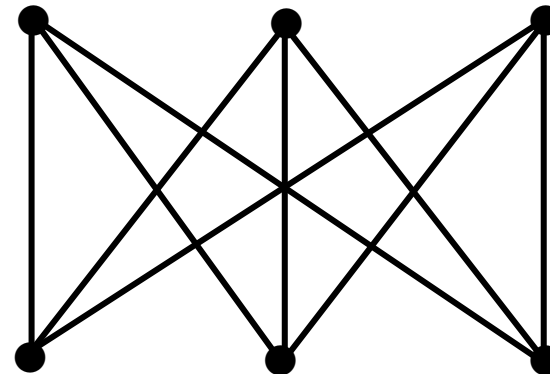
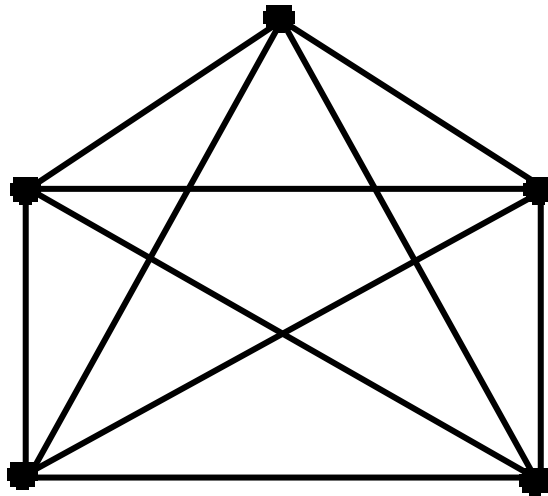
5. Graphes planaires



- Un graphe non orienté est **planaire** s'il est possible de le dessiner dans le plan de telle sorte que deux arêtes quelconques ne se rencontrent pas en dehors de leurs extrémités.

Graphes planaires

- Applications : réalisation de circuits imprimés, de microprocesseurs, ...
- Exemples de graphes non-planaires :



Chemins les plus courts et les plus longs dans un graphe valué

- Graphe orienté valué.
 - Valeurs : longueurs, coûts, durées, délais, ...
- Problème : Chercher un chemin d'un sommet x_1 vers un sommet x_n de telle sorte que la somme des valeurs des arcs qui composent ce chemin ('longueur du chemin') soit minimum (ou maximum).

Hypothèses

- Pour un problème à maximum, pas de circuits de valeur positive.
- Pour un problème à minimum, pas de circuits de valeur négative.
- Notations :
 - arc $(x_i, x_j) \leftrightarrow$ valeur c_{ij}
 - Pour un problème à maximum :
$$\begin{cases} c_{ij} = -\infty & \text{si } (x_i, x_j) \notin U \text{ et } i \neq j \\ c_{ii} = 0 & \forall i \end{cases}$$
 - Pour un problème à minimum :
$$\begin{cases} c_{ij} = +\infty & \text{si } (x_i, x_j) \notin U \text{ et } i \neq j \\ c_{ii} = 0 & \forall i \end{cases}$$
 - Passage de minimum à maximum : $c'_{ij} = -c_{ij} \quad , \quad \forall i, j$

6. Algorithme de Ford

- Marquage des sommets : $x_i \rightarrow \lambda_i$
- 1. On pose $\lambda_1 = 0$ et $\lambda_i = +\infty$, $\forall i \neq 1$
- 2. On cherche (x_i, x_j) tel que $\lambda_j - \lambda_i > c_{ij}$
et on remplace λ_j par $\lambda'_j = \lambda_i + c_{ij}$
- 3. On continue ainsi tant qu'il est possible
de diminuer des λ_i

Algorithme de Ford

- Obtention du chemin le plus court de x_1 à x_n :
 - on cherche x_{p1} tel que : $\lambda_n - \lambda_{p1} = c_{p1pn}$,
 - on cherche x_{p2} tel que : $\lambda_{p1} - \lambda_{p2} = c_{p2p1}$,Et ainsi de suite jusqu'à trouver : $x_{pk} = x_1$
- Convergence : l'algorithme se termine en au plus m étapes ($m =$ nombre d'arcs).

Théorème

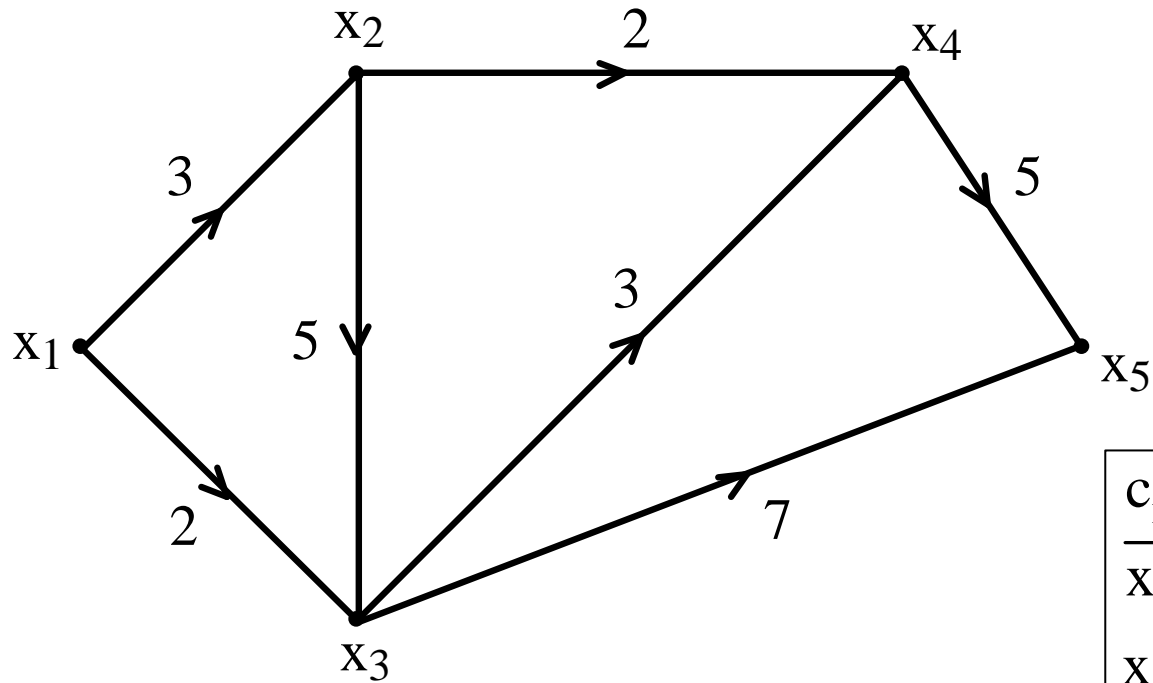
Le chemin (x_1 , $x_{p_{k-1}}$, $x_{p_{k-2}}$, \dots , x_{p_1} , x_n), de valeur λ_n , est optimal.

Démonstration :

Soit (x_1 , x_{q_1} , x_{q_2} , \dots , x_{q_r} , x_n) un chemin quelconque de valeur Q :

$$\begin{aligned} \lambda_{q_1} - \lambda_1 &\leq c_{1q_1} \\ \lambda_{q_2} - \lambda_{q_1} &\leq c_{q_1q_2} \\ &\vdots \\ \lambda_n - \lambda_{q_r} &\leq c_{q_r n} \\ \hline \lambda_n &\leq Q \end{aligned}$$

Exemple 5



c_{ij}	x_1	x_2	x_3	x_4	x_5
x_1	0	3	2	∞	∞
x_2	∞	0	5	2	∞
x_3	∞	∞	0	3	7
x_4	∞	∞	∞	0	5
x_5	∞	∞	∞	∞	0

Résolution dans Excel (1)

$$0+3 = 3 \quad 0+2 = 2$$

	Sommets :	1	2	3	4	5
	Lambda :	0	10000	10000	10000	10000
Lambda						
0	cij	0	3	2	10000	10000
10000		10000	0	5	2	10000
10000		10000	10000	0	3	7
10000		10000	10000	10000	0	5
10000		10000	10000	10000	10000	0
	Delta	0	9997	9998	0	0
		-20000	0	-5	-2	-10000
		-20000	-10000	0	-3	-7
		-20000	-10000	-10000	0	-5
		-20000	-10000	-10000	-10000	0

$$(\Delta = \lambda_j - \lambda_i - c_{ij})$$

Résolution dans Excel (2)

$$3+2 = 5$$

	Sommets :	1	2	3	4	5
	Lambda :	0	3	2	10000	10000
Lambda						
0	cij	0	3	2	10000	10000
3		10000	0	5	2	10000
2		10000	10000	0	3	7
10000		10000	10000	10000	0	5
10000		10000	10000	10000	10000	0
	Delta	0	0	0	0	0
		-10003	0	-6	9995	-3
		-10002	-9999	0	9995	9991
		-20000	-19997	-19998	0	-5
		-20000	-19997	-19998	-10000	0

$$(\Delta = \lambda_j - \lambda_i - c_{ij})$$

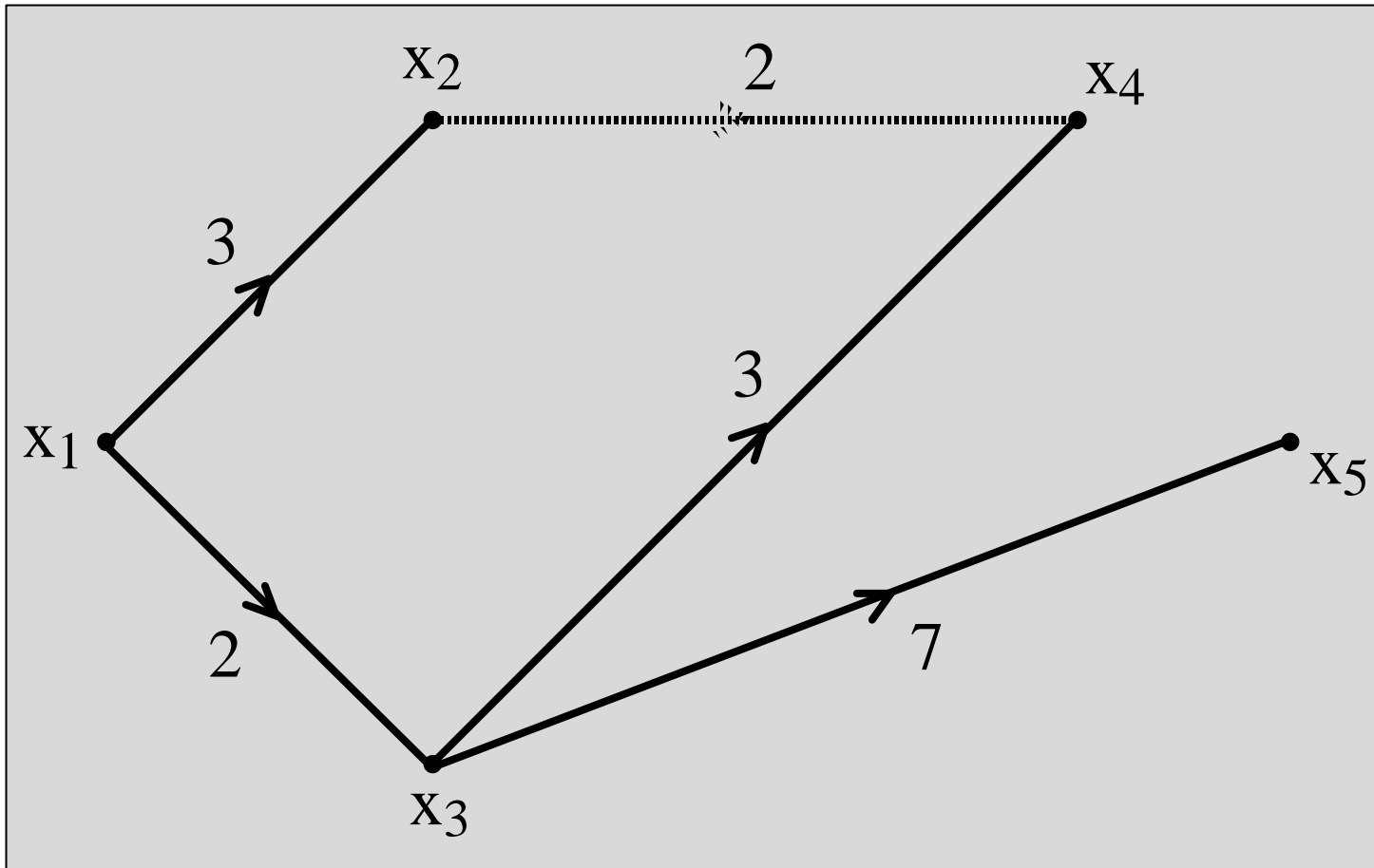
Résolution dans Excel (3)

$$2+7 = 9$$

	Sommets :	1	2	3	4	5
	Lambda :	0	3	2	5	10000
Lambda						
0	cij	0	3	2	10000	10000
3		10000	0	5	2	10000
2		10000	10000	0	3	7
5		10000	10000	10000	0	5
10000		10000	10000	10000	10000	0
	Delta	0	0	0	-9995	0
		-10003	0	-6	0	-3
		-10002	-9999	0	0	9991
		-10005	-10002	-10003	0	9990
		-20000	-19997	-19998	-19995	0

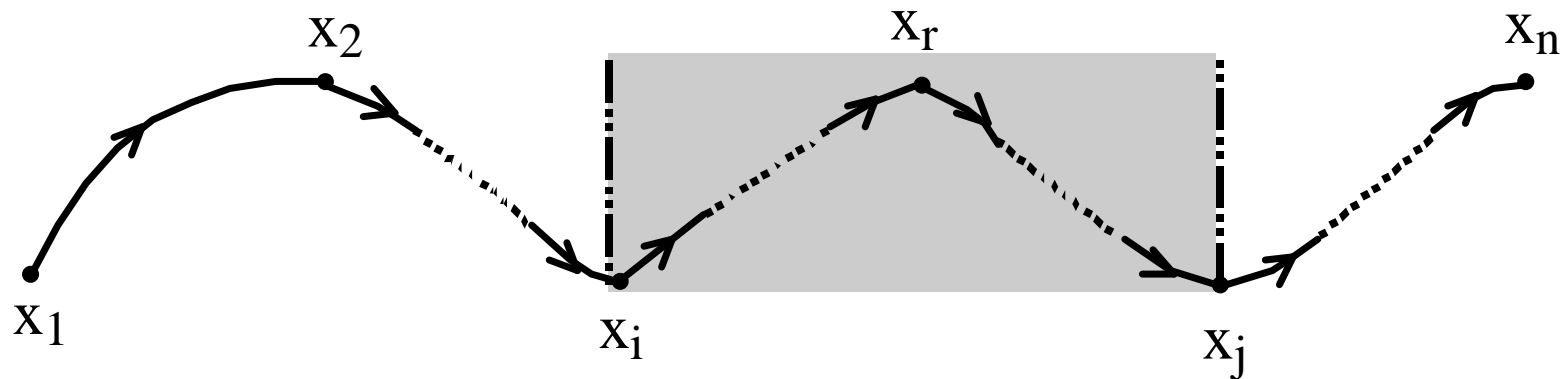
$$(\Delta = \lambda_j - \lambda_i - c_{ij})$$

Arborescence des chemins les plus courts



7. Algorithme de Bellman-Kalaba

- Principe d'optimalité de Bellman :



- Si le chemin $(x_1, x_2, \dots, x_i, \dots, x_r, \dots, x_j, \dots, x_n)$ est optimal entre x_1 et x_n , alors le chemin $(x_i, \dots, x_r, \dots, x_j)$ est optimal entre x_i et x_j .

Algorithme de Bellman-Kalaba

- $\lambda_i(k)$ = valeur du chemin optimal de x_1 à x_i en au plus k arcs.

1.
$$\begin{cases} \lambda_1(1) = 0 \\ \lambda_i(1) = c_{1i} \quad i = 2, \dots, n \end{cases}$$

2. Pour $k = 2, 3, \dots$

$$\begin{cases} \lambda_1(k) = 0 \\ \lambda_i(k) = \min_{j \in \{1, 2, \dots, n\}} \{ \lambda_j(k-1) + c_{ji} \} \end{cases}$$

3. Arrêt de l'algorithme quand :

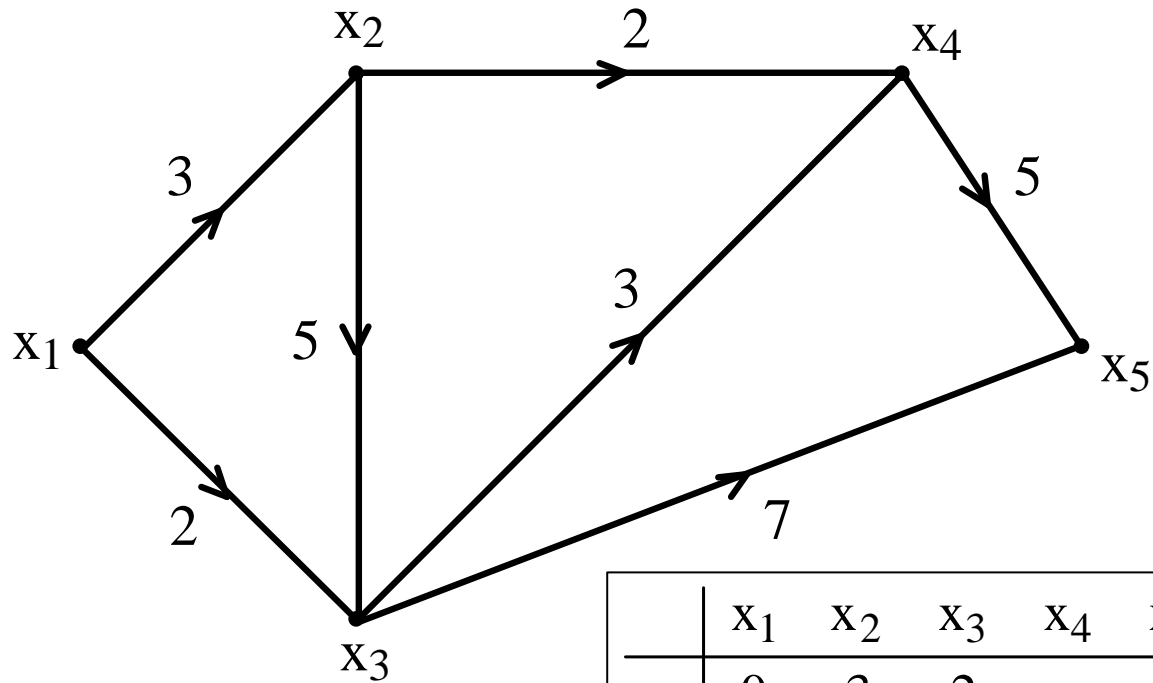
$$\lambda_i(k) = \lambda_i(k-1) \quad \forall i$$

Algorithme de Bellman-Kalaba

- Convergence :
Arrêt de l'algorithme après au plus $n-1$ étapes.
- Recherche de chemins les plus longs :
Adaptation du point 2 :

$$\lambda_i(k) = \max_j \{ \lambda_j(k-1) + c_{ji} \}$$

Exemple 5



	x_1	x_2	x_3	x_4	x_5	$\lambda_i(1)$	$\lambda_i(2)$	$\lambda_i(3)$
x_1	0	3	2	∞	∞			
x_2	∞	0	5	2	∞			
x_3	∞	∞	0	3	7			
x_4	∞	∞	∞	0	5			
x_5	∞	∞	∞	∞	0			

8. Algorithme de Dijkstra

- Uniquement pour des valeurs positives !
- Terminologie :
 - $\lambda_i(k)$: marque du sommet i à l'étape k ,
 - λ_i° : marque définitive du sommet i ,
 - $M(k)$: ensemble des sommets marqués définitivement à la fin de l'étape k ,
 - n_k : numéro du sommet marqué définitivement à l'étape k .

Algorithme de Dijkstra

1. Etape 1 :
(initialisation)

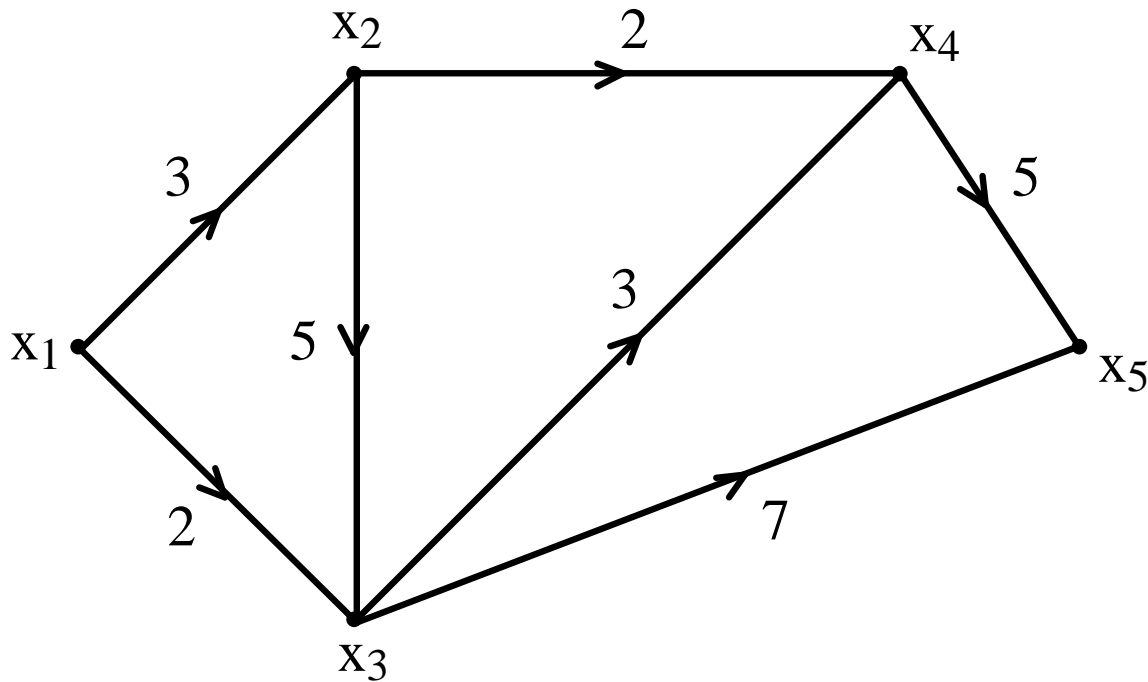
$$\left\{ \begin{array}{l} \lambda_i(1) = \infty, \quad \forall i \neq 1 \\ \lambda_1(1) = 0 = \lambda_1^\circ \\ n_1 = 1 \\ M(1) = \{1\} \end{array} \right.$$
2. Etape $k+1$:
($k = 1, 2, \dots$)

$$\left\{ \begin{array}{l} \lambda_i(k+1) = \min \left\{ \lambda_i(k), \lambda_{n_k}^\circ + c_{n_k i} \right\} \quad \forall i \notin M(k) \\ \lambda_{n_{k+1}}^\circ = \min_i \lambda_i(k+1) \\ M(k+1) = M(k) \cup \{n_{k+1}\} \end{array} \right.$$

Algorithme de Dijkstra

- Fin de l'algorithme quand tous les sommets sont définitivement marqués.
- Convergence : en au maximum n étapes.
- Propositions :
 - $\lambda_i(k)$ = valeur du chemin optimal de x_1 à x_i ne passant que par des sommets de $M(k-1)$.
 - λ_{nk}° = valeur du chemin optimal de x_1 à x_{nk} (ce chemin ne passe que par des sommets de $M(k-1)$).

Exemple 5

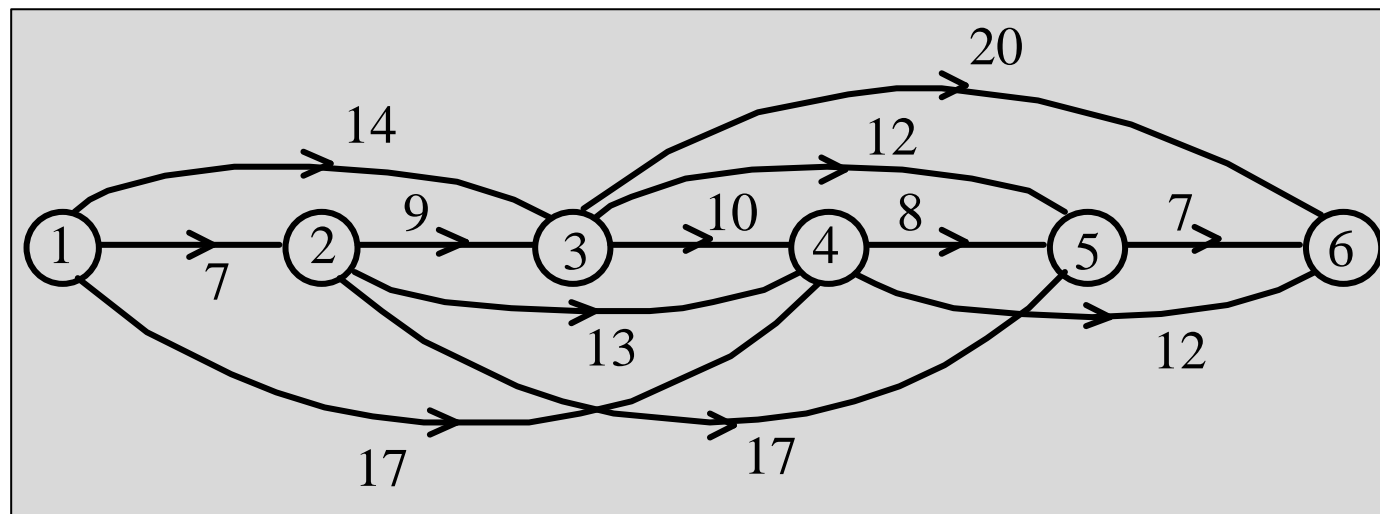


	x_1	x_2	x_3	x_4	x_5	
x_1	0	3	2	∞	∞	
x_2	∞	0	5	2	∞	
x_3	∞	∞	0	3	7	
x_4	∞	∞	∞	0	5	
x_5	∞	∞	∞	∞	0	
	0	3	2	∞	∞	$\lambda_i(1)$
		3	2	∞	∞	$\lambda_i(2)$
		3		5	9	$\lambda_i(3)$
				5	9	$\lambda_i(4)$
					9	$\lambda_i(5)$

Application

- Renouvellement d'un équipement : optimisation de la politique d'achat sur 5 ans.

C_{ij} = coût d'un équipement acheté au début de l'année i et revendu au début de l'année j .



Plan du cours

1. Introduction
 - Historique, modélisation
2. Quelques problèmes de la théorie des graphes
 - Définitions, terminologie
 - Arbre partiel minimum, Coloration
 - Chemins les plus courts et les plus longs
3. Problèmes d'ordonnancement
 - Méthode du chemin critique
 - Contraintes cumulatives
 - Méthode PERT
4. Réseaux de transport
 - Flot maximum et coupe minimum
 - Cas particuliers et variantes
5. Problème du voyageur de commerce

A. Définition du problème

- Réalisation en un temps minimum d'un projet comportant un certain nombre de tâches à effectuer, en tenant compte de contraintes éventuelles sur l'enchaînement des tâches ou sur les moyens à mettre en oeuvre.
- Exemples :
 - Construction d'une maison, chantier, campagne publicitaire, lancement d'un nouveau produit, ...

Ordonnancement

- Déterminer la date de début de chaque tâche (c-à-d un ordonnancement).
- Notations :
 - Tâches : $i \quad i = 1, 2, \dots, n$
 - Durées : $d(i)$
 - Dates de début : $t(i)$
- Prise en compte de contraintes temporelles et cumulatives.

Contraintes temporelles

- Postériorité stricte $t(j) \geq t(i) + d(i)$
- Postériorité avec délai $t(j) \geq t(i) + d(i) + f(i, j)$
- Postériorité partielle $t(j) \geq t(i) + \alpha(i, j)d(i)$
- Localisation temporelle $t(i) \geq a(i)$
- Continuité $t(j) \leq t(i) + t(i, j)$

Contraintes cumulatives

- Limites sur les ressources disponibles pendant la réalisation du projet :
 - Matériel,
 - Budget,
 - Main-d'œuvre.
- Fixes ou variables au cours du temps.

Exemple - Exploitation minière

- En vue de l'exploitation d'une mine, on construit :
 - un port sur un canal proche du site d'extraction,
 - ainsi qu'une route et une voie de chemin de fer qui relie la mine au port.
- Au total, il y a 10 tâches à effectuer (durées exprimées en mois).
- Contraintes temporelles uniquement (dans un premier temps).

Exemple - Liste des tâches

Tâches	Durées	Contraintes
A: Construction d'un port provisoire	2	-
B: Déblai pour route et voie ferrée	3	-
C: Commande du matériel minier	5	-
D: Commande du matériel portuaire	6	-
E: Implantation du port définitif	2	Après A.
F: Construction de la route	4	Après B.
G: Pose de la voie ferrée	1	Après B.
H: Installation portuaire	8	Après D.
I : Construction d'une cité	5	Après A,B,E,F.
J: Installation minière	7	Après B,C,D,G,H.

Exemple - Questions

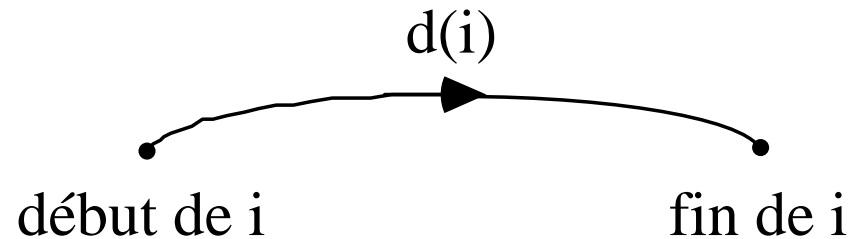
- Quelle est la durée de réalisation minimale des travaux, en tenant compte des contraintes ?
- Quel est le calendrier d'exécution des différentes tâches correspondant ?
- On pourrait aussi prendre en compte le nombre d'ouvriers disponibles et leurs qualifications, ...

B. Méthode du chemin critique

- Contraintes temporelles uniquement.
- Durées des tâches connues avec certitude.
- Représentation sous forme de graphe valué :
 - Tâches représentées par des arcs.
 - Sommets correspondant à des étapes du projet.

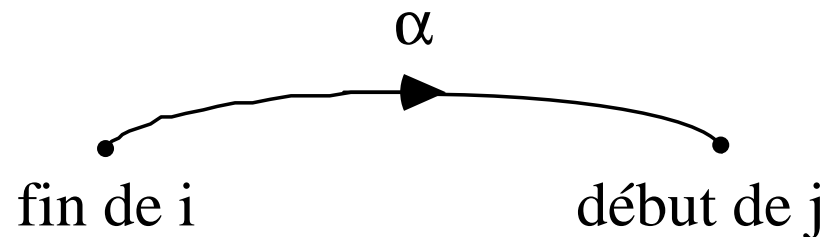
Elaboration du graphe

- Tâche i :



- Contrainte :

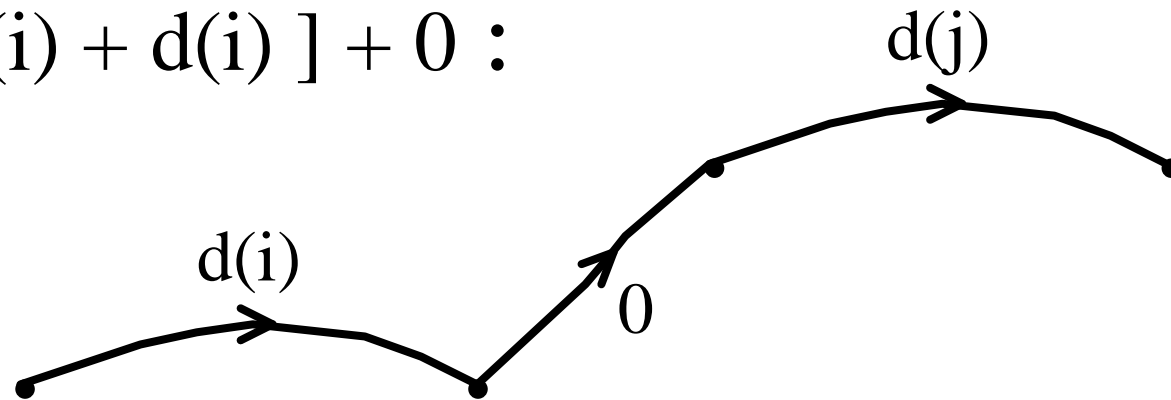
$$t(j) \geq t(i) + d(i) + \alpha$$



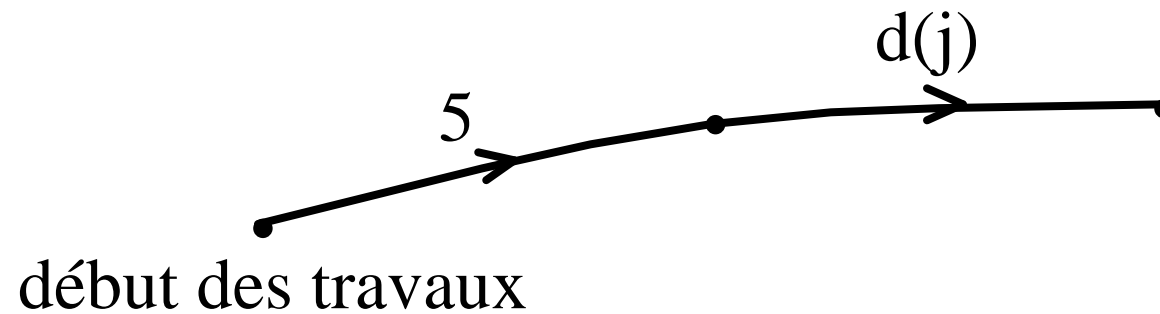
- Deux sommets particuliers :
 - Début du projet - étape 0,
 - Fin du projet - étape $n+1$.

Exemples

- $t(j) \geq [t(i) + d(i)] + 0$:



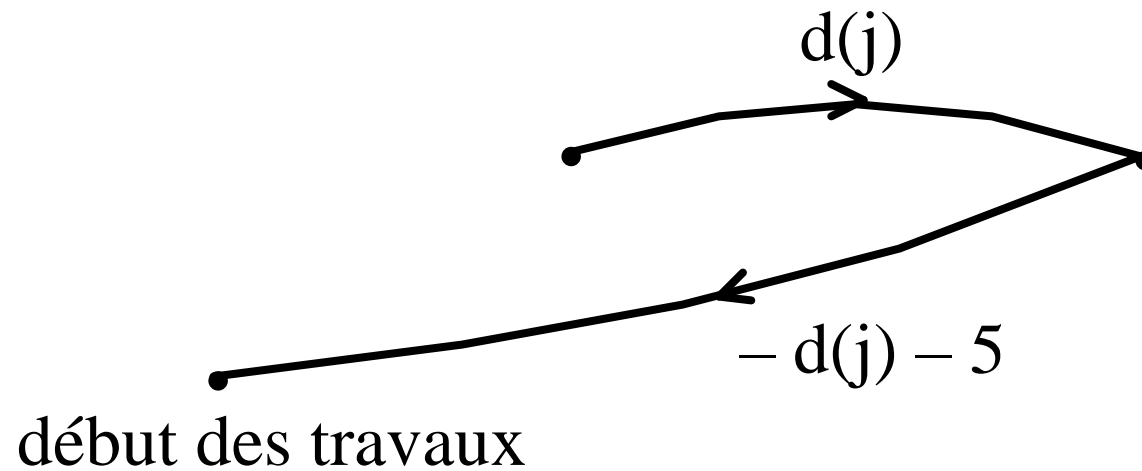
- $t(j) \geq [t(0) + 0] + 5$:



Exemples

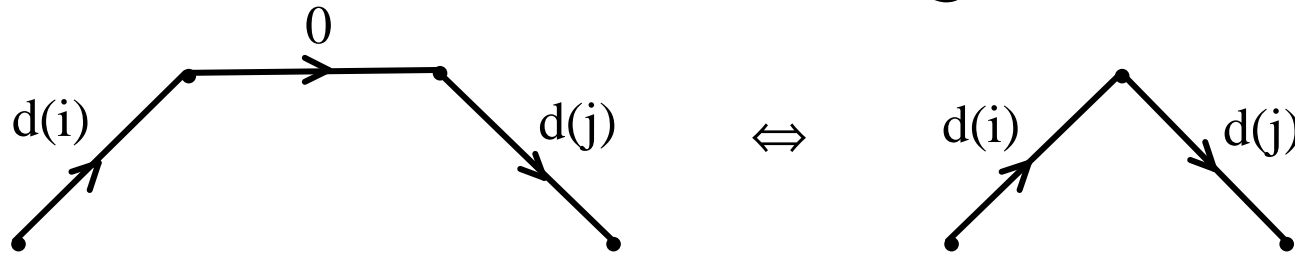
- $t(j) \leq t(0) + 5$:

$$t(0) \geq [t(j) + d(j)] - d(j) - 5$$

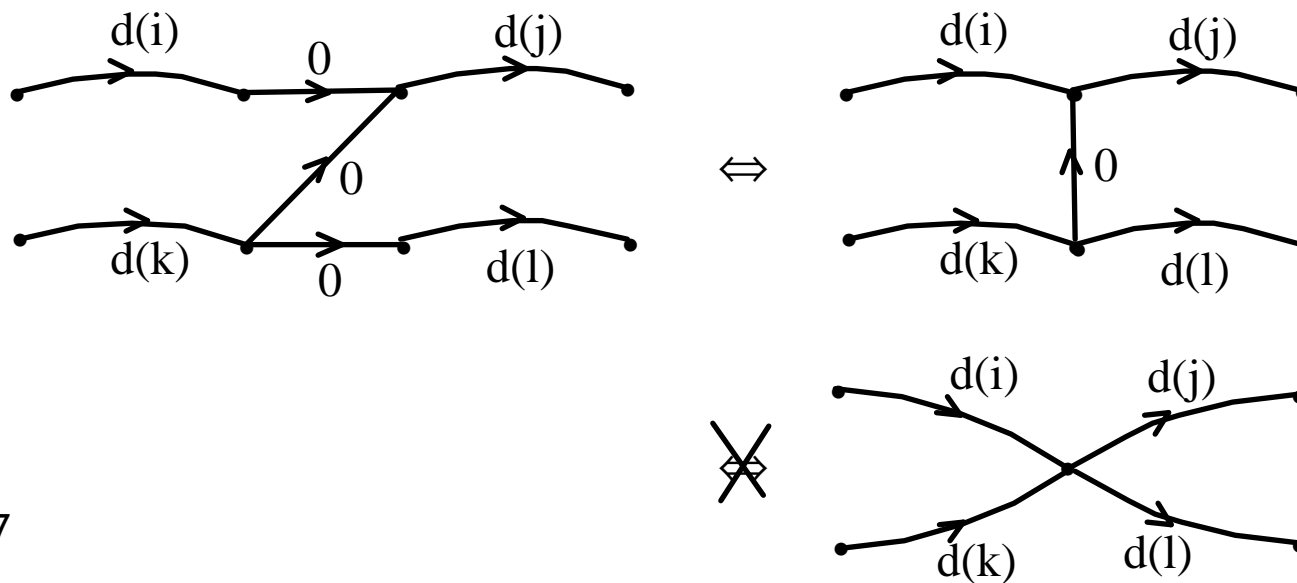


Simplification du graphe

- Elimination des arcs de longueur nulle :



- Attention :



Ordonnancement au plus tôt

- **ES(i)** = date de début au plus tôt de la tâche i
= longueur du chemin le plus long de 0 au début de i
 - ▶ $ES(n+1) = T$ = durée minimale de réalisation du projet.
- **EF(i)** = date de fin au plus tôt de la tâche $i = ES(i) + d(i)$

Ordonnancement au plus tard

- **LF(i)** = date de fin au plus tard de la tâche i
(*sans allonger la durée de réalisation T*)
= T – valeur du chemin de valeur maximum de la fin de i à la fin des travaux
- **LS(i)** = date de début au plus tard de la tâche i = $LF(i) - d(i)$

Calcul

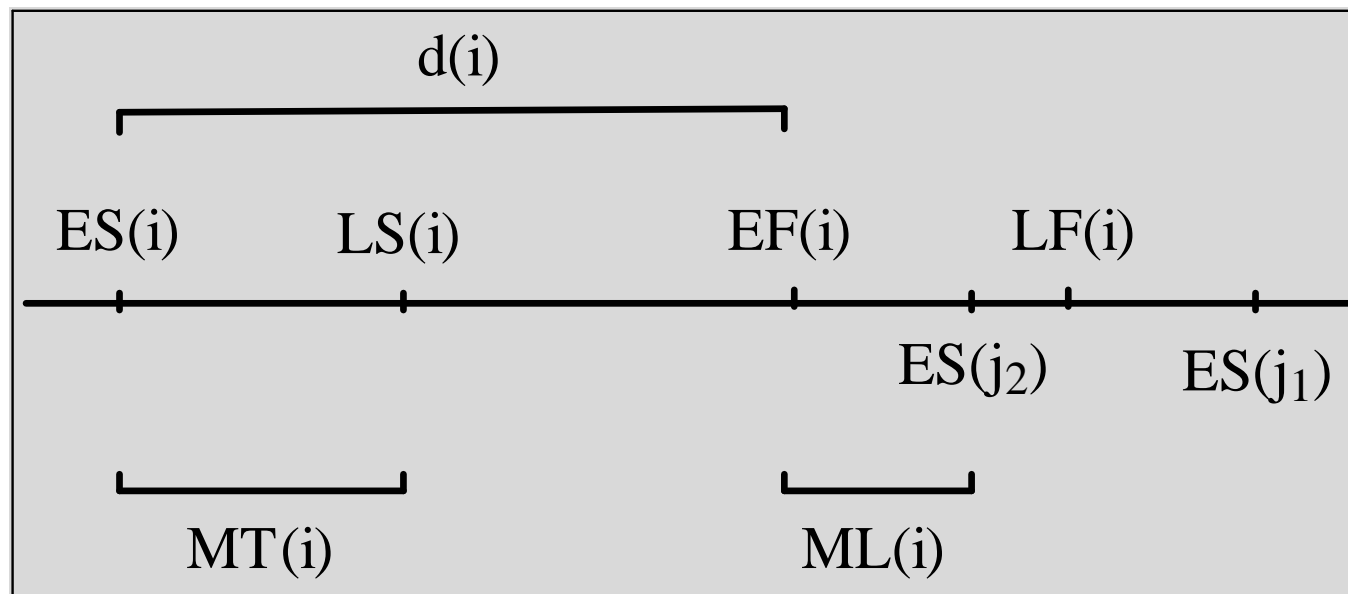
- Forme simplifiée de l'algorithme de Bellman-Kalaba pour un graphe sans circuits.
- Classement des sommets en k niveaux.
- Calcul des dates de début au plus tôt pour les sommets, par niveau décroissant, depuis le début des travaux :
 - $ES(0) = 0$
 - ensuite : $ES(i) = \max \{ ES(j) + c_{ji} \mid j \in \Gamma^-(i) \}$
- Puis calcul des dates au plus tard en repartant de la fin des travaux.

Chemin critique

- Le chemin critique est le chemin le plus long entre le début et la fin des travaux.
- Tâches critiques :
 - Tâches situées sur le chemin critique.
 - Tout retard sur une tâche critique rallonge d'autant la durée minimale de réalisation du projet.

Marges

- Retards possibles sur les tâches non-critiques, sans augmenter T ?



Marges

- Marge totale de la tâche i :

$$MT(i) = LS(i) - ES(i)$$

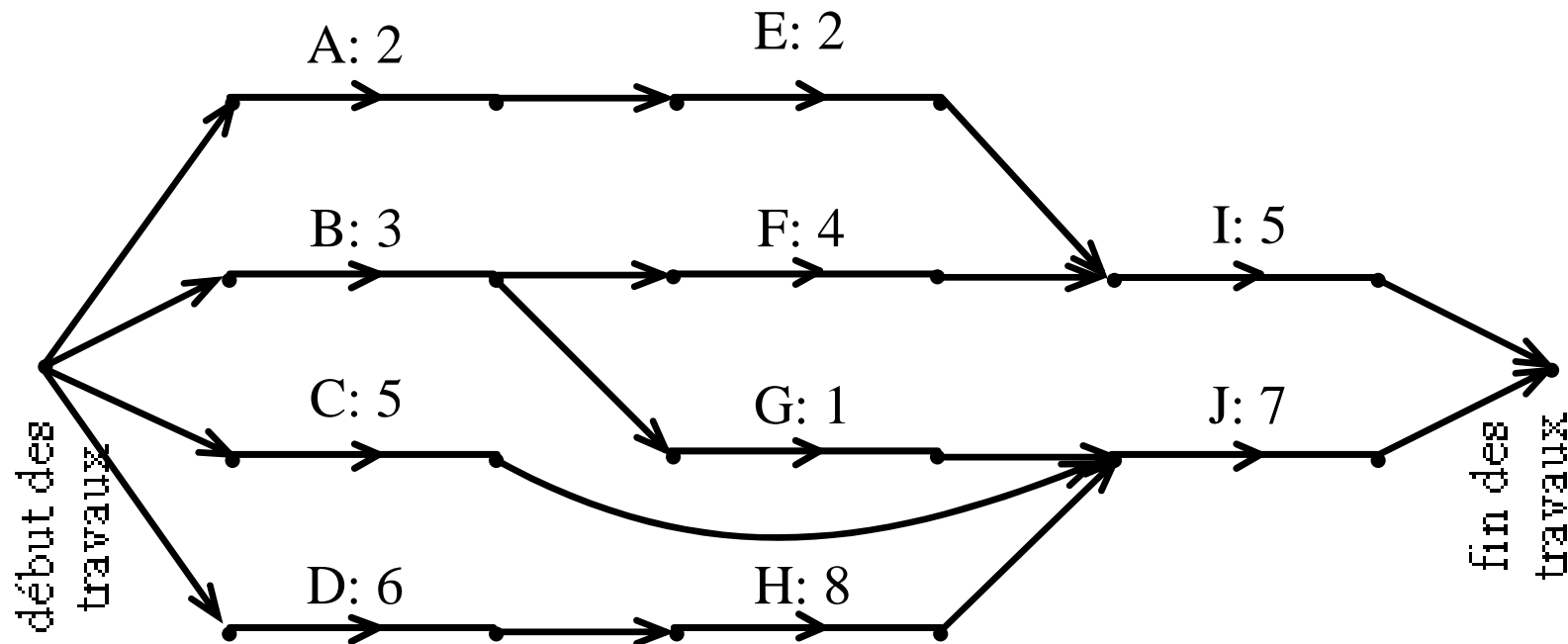
- Retard maximum sans augmenter T .

- Marge libre de la tâche i :

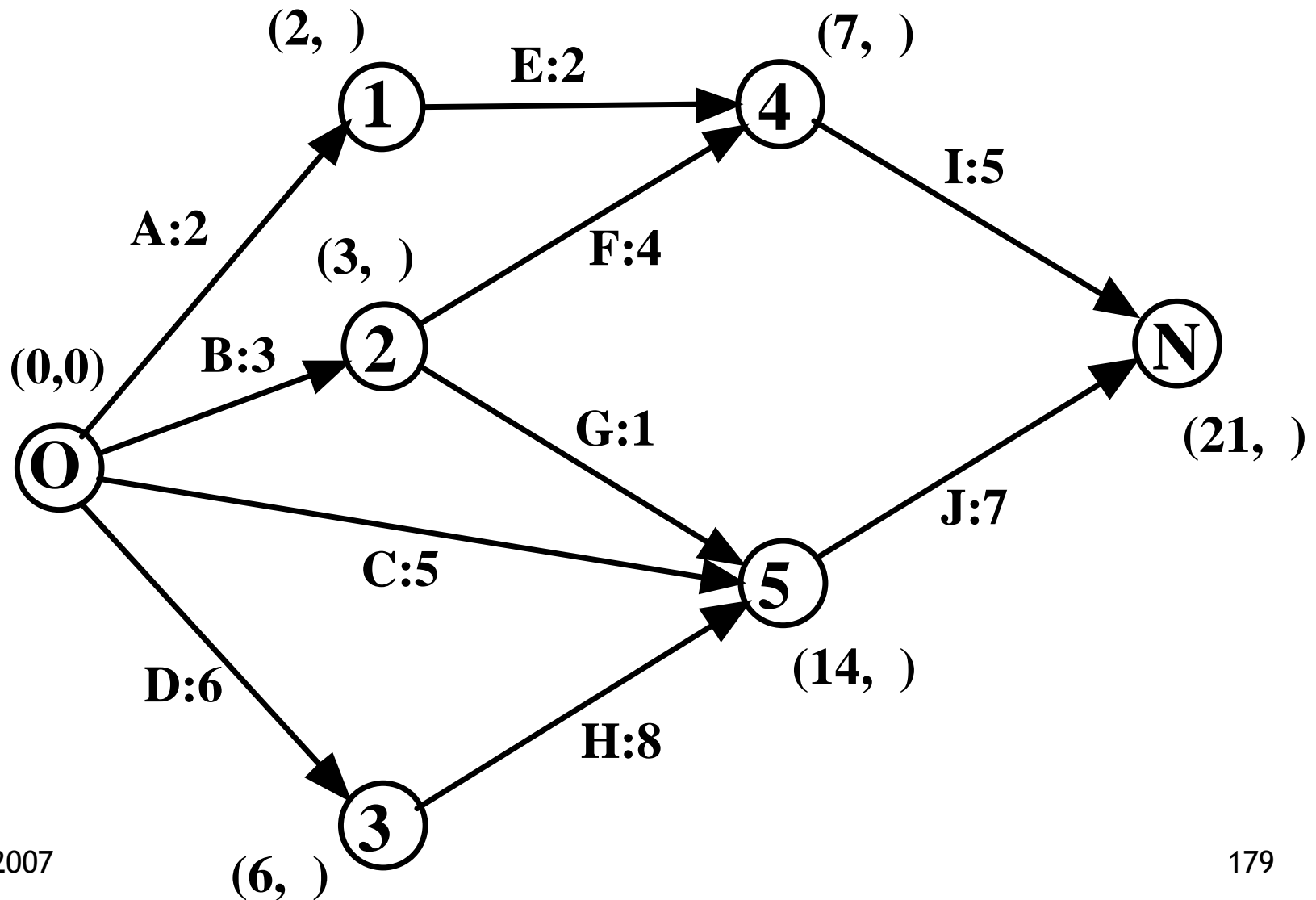
$$ML(i) = \min \{ ES(j) - EF(i) \mid j \text{ suit } i \}$$

- Retard maximum sans perturber les dates au plus tôt.

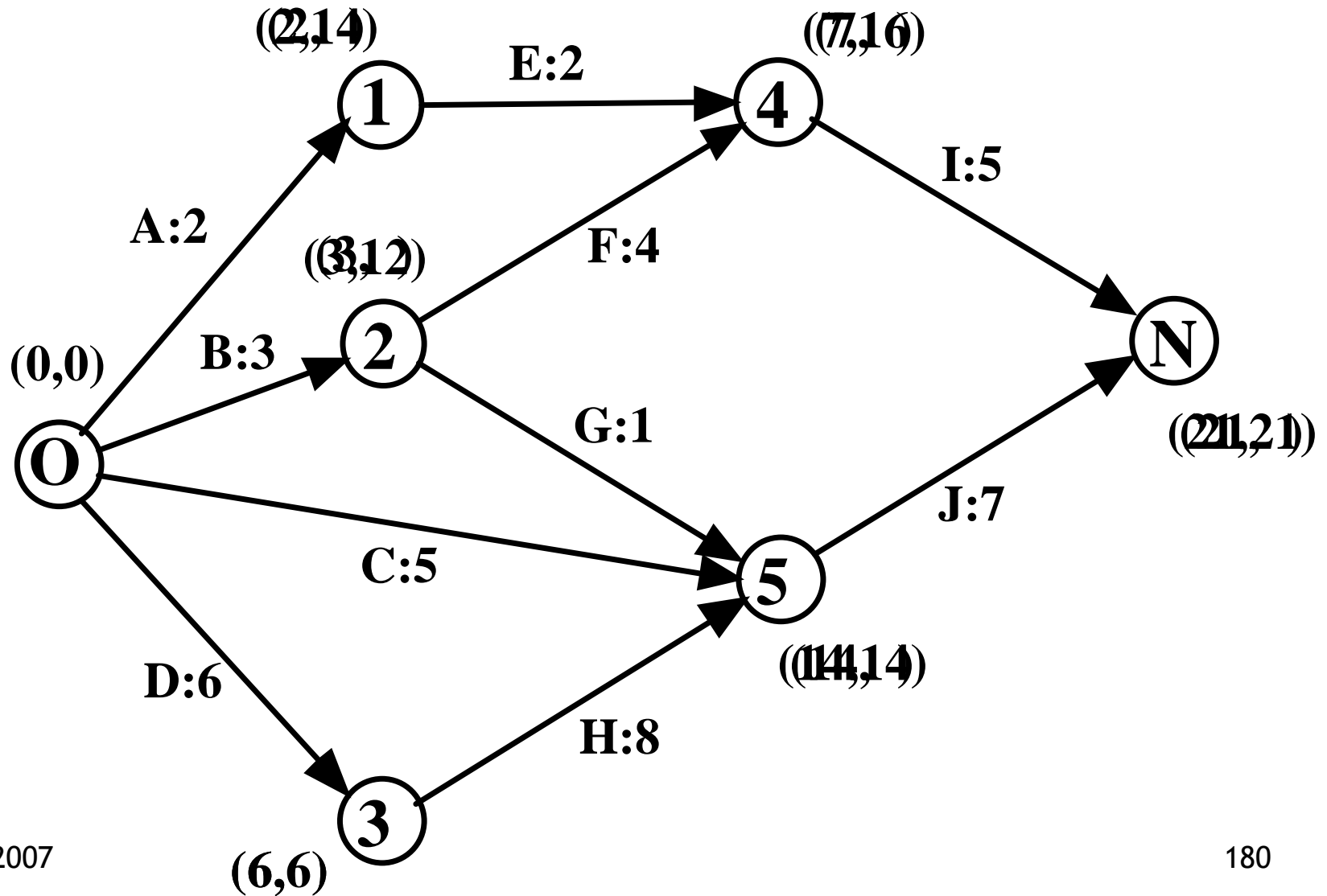
Exemple - graphe



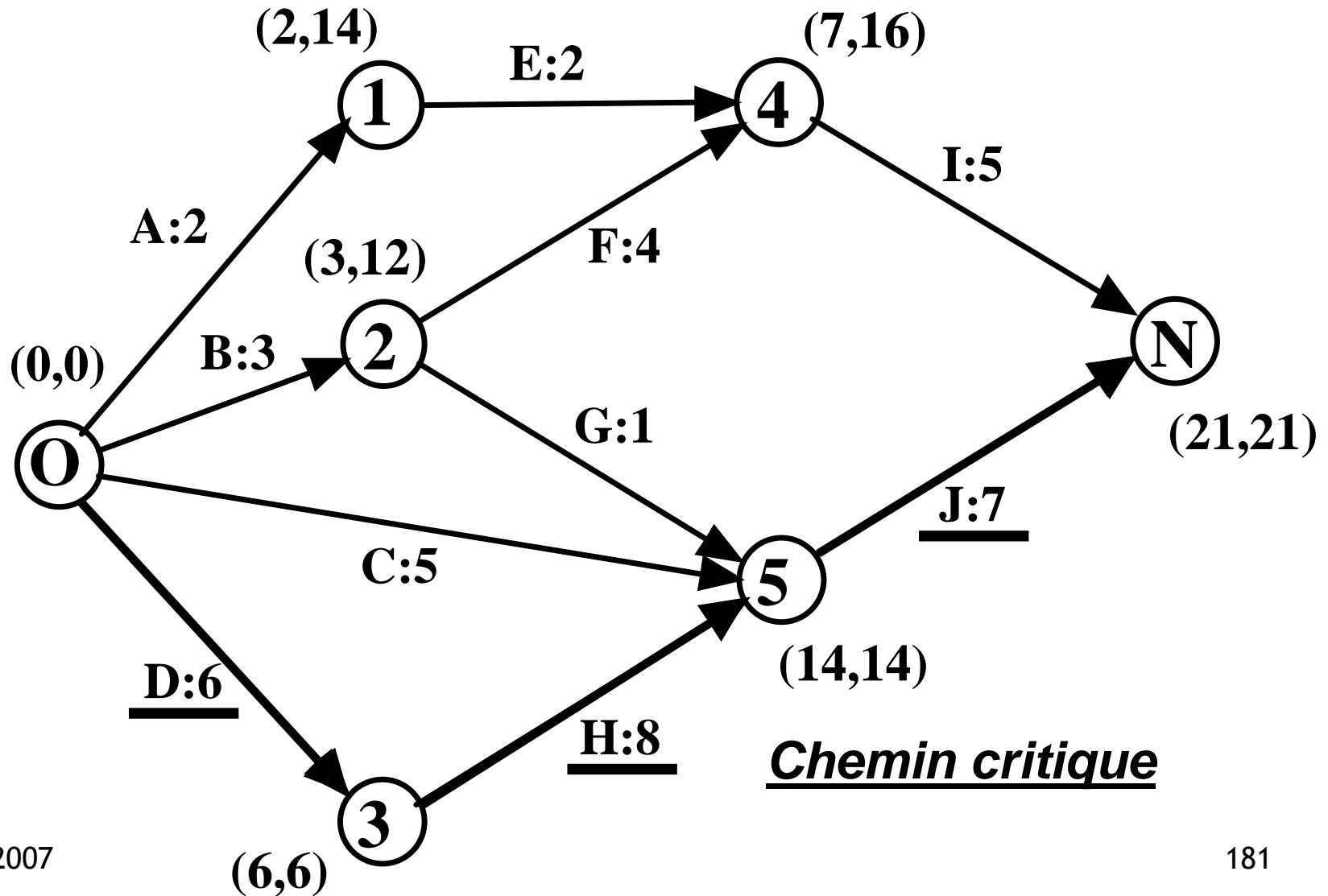
Exemple - graphe simplifié



Exemple - graphe simplifié



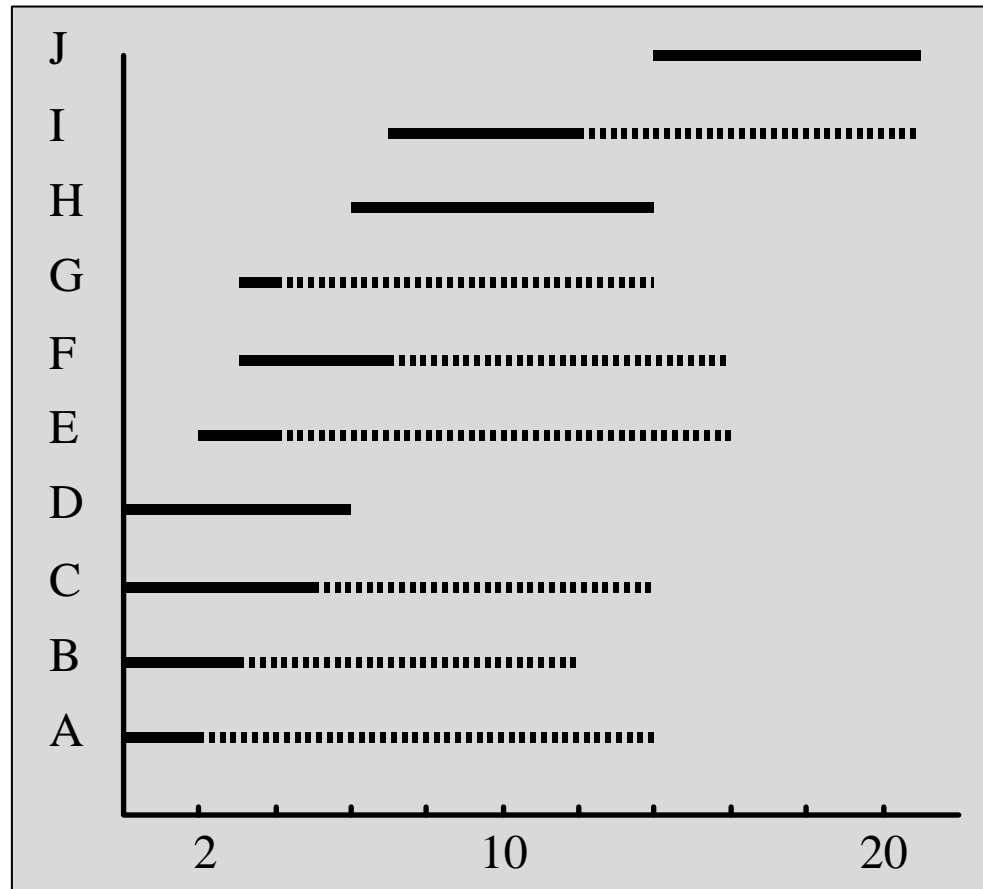
Exemple - graphe simplifié



Exemple - résultats

Tâche	Durée	Après	ES	EF	LS	LF	ML	MT
A	2	-	0	2	12	14	0	12
B	3	-	0	3	9	12	0	9
C	5	-	0	5	9	14	9	9
D	6	-	0	6	0	6	0	0
E	2	A	2	4	14	16	3	12
F	4	B	3	7	12	16	0	9
G	1	B	3	4	13	14	10	10
H	8	D	6	14	6	14	0	0
I	5	A,B,E,F	7	12	16	21	9	9
J	7	B,C,D,G,H	14	21	14	21	0	0

Diagramme de Gantt



- Trait plein :
 - Ordonnancement au plus tôt.
- Pointillés :
 - Marge totale.

Contraintes cumulatives

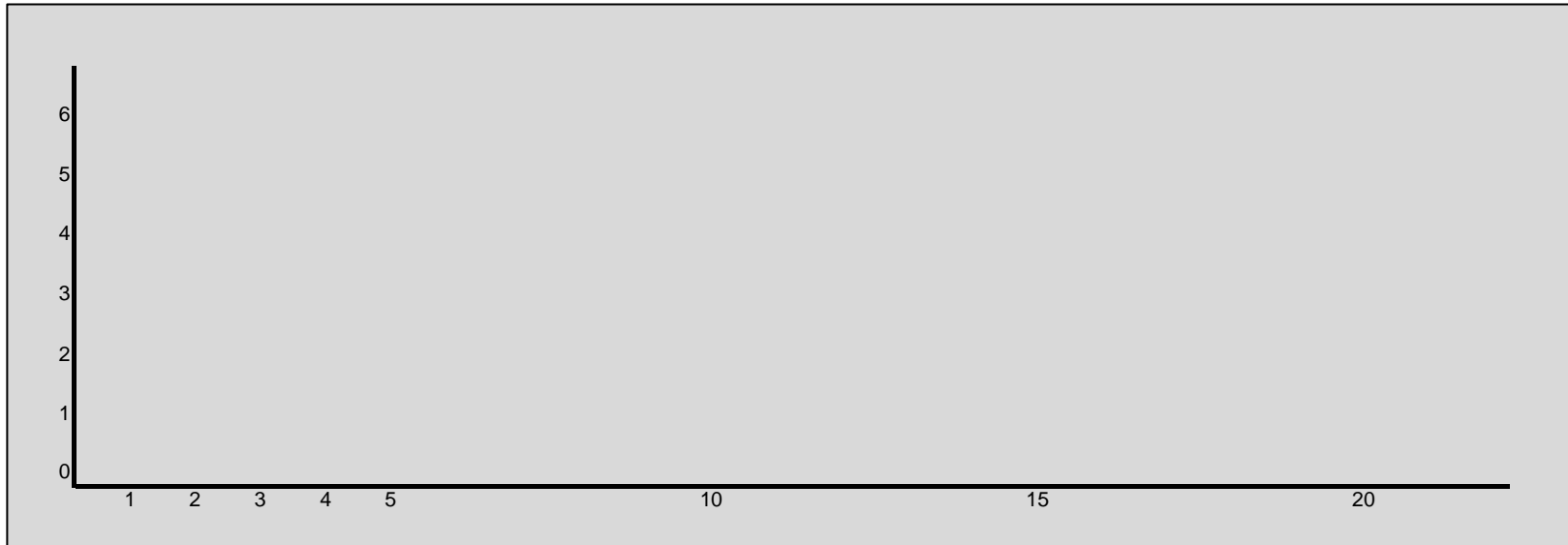
- Origine :
 - Utilisation de ressources disponibles en quantités limitées :
outils, matières premières, ouvriers, budget, ...
- Courbe de charge :
 - Pour un type de ressource et pour un ordonnancement donné : quantités cumulées nécessaires en fonction du temps.
- Problèmes :
 - Respecter un profil maximum pour la courbe de charge (contrainte).
 - Lisser la courbe de charge (éviter les pics).

Exemple

- Ressource : 3 équipes d'ouvriers disponibles.
- Equipes nécessaires par tâche :
(pendant toute la durée de la tâche)

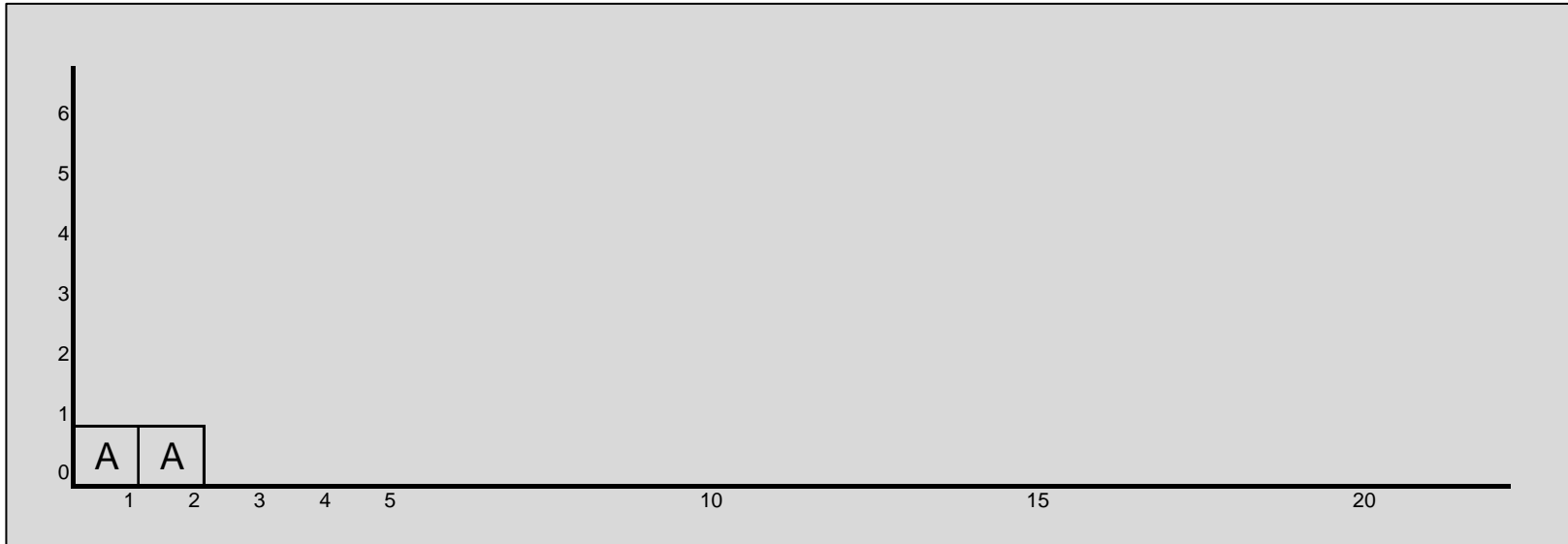
A : 1	B : 1	C : 0	D : 0	E : 3
F : 2	G : 1	H : 0	I : 3	J : 1

Courbe de charge



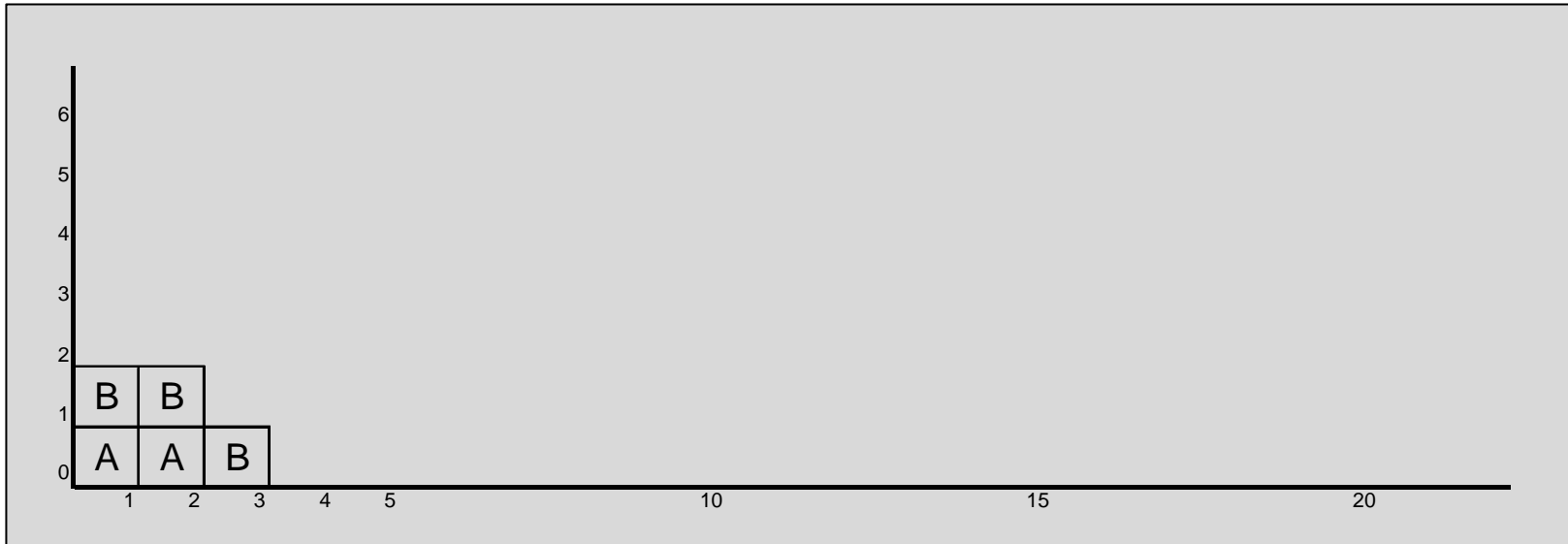
- Tâche A :
 - Durée : 2 mois,
 - Dès le début des travaux,
 - 1 équipe d'ouvriers.

Courbe de charge



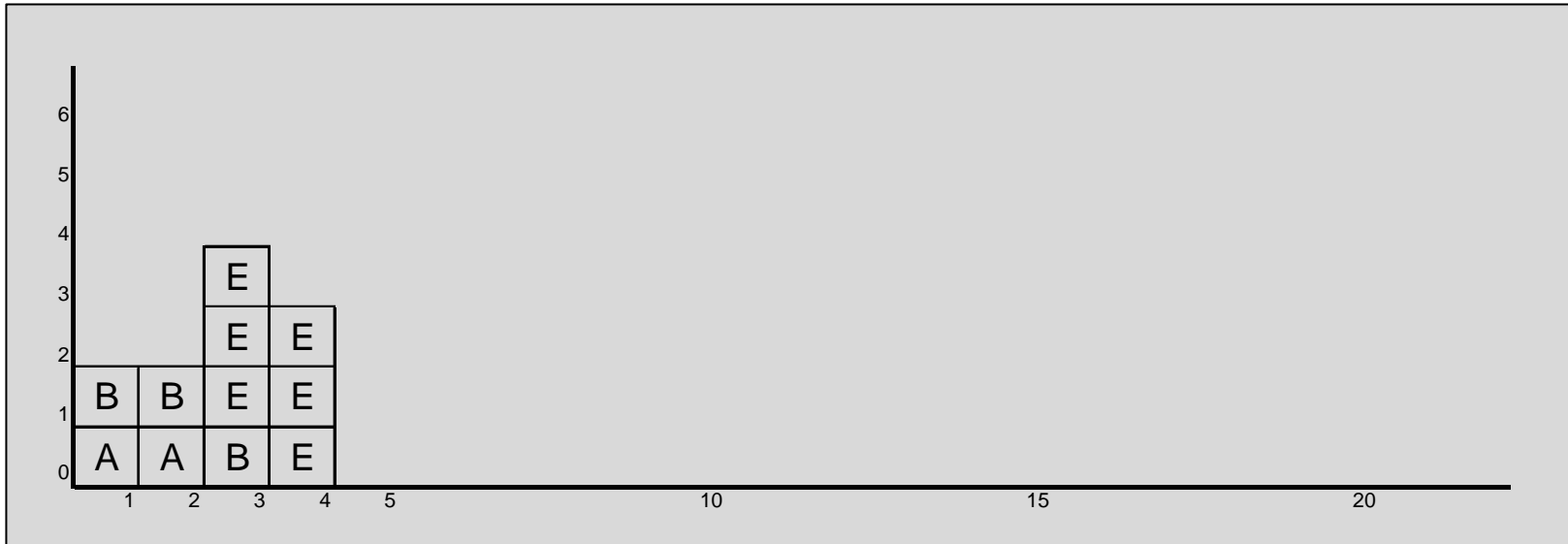
- Tâche B :
 - Durée : 3 mois,
 - Dès le début des travaux,
 - 1 équipe d'ouvriers.

Courbe de charge



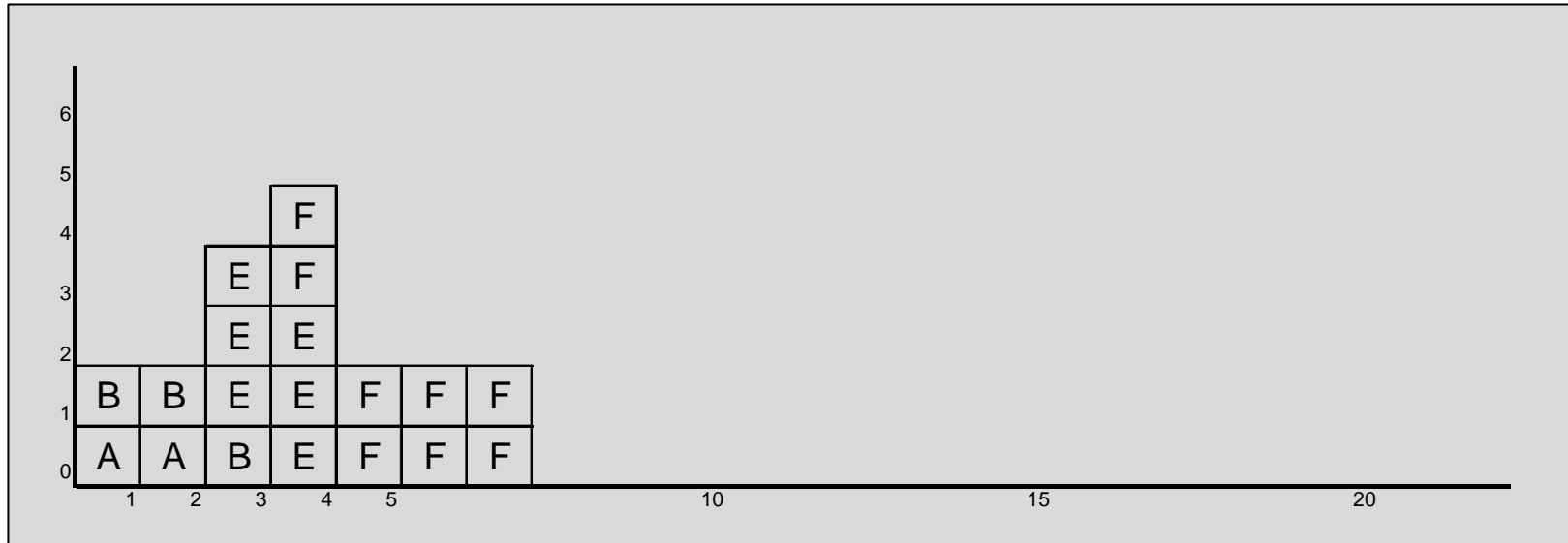
- Tâche E :
 - Durée : 2 mois,
 - Au plus tôt 2 mois après le début des travaux,
 - 3 équipes d'ouvriers.

Courbe de charge



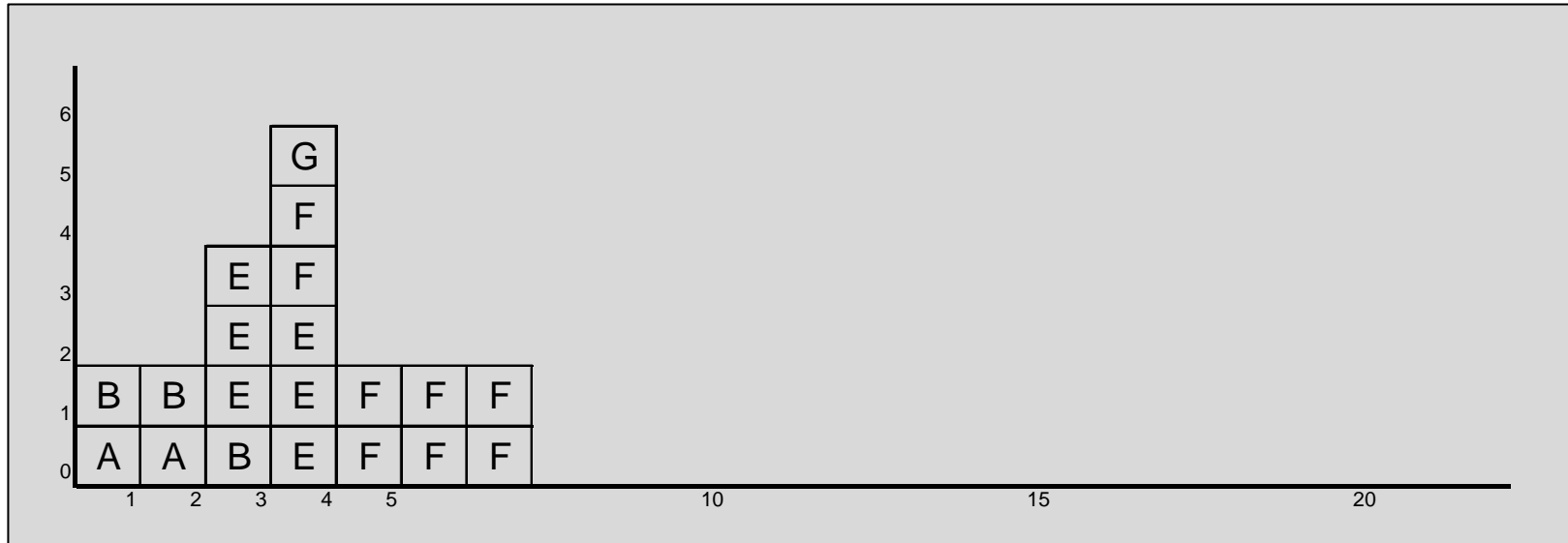
- Tâche F :
 - Durée : 4 mois,
 - Au plus tôt 3 mois après le début des travaux,
 - 2 équipes d'ouvriers.

Courbe de charge



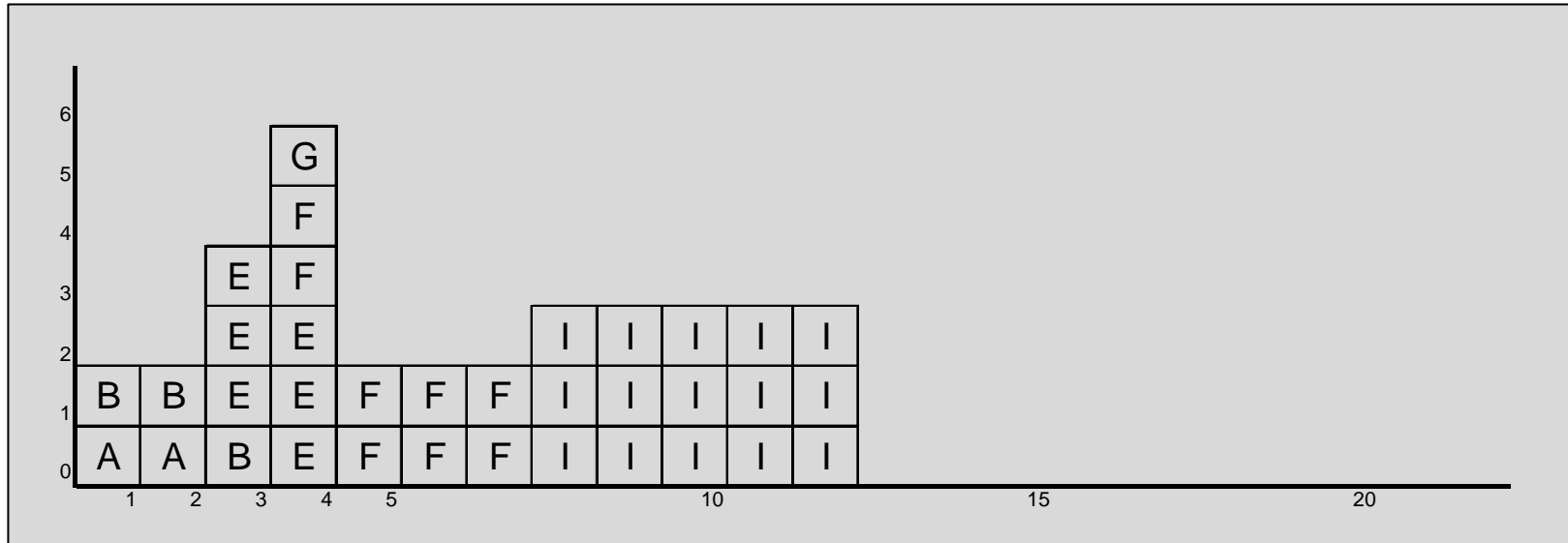
- Tâche G :
 - Durée : 1 mois,
 - Au plus tôt 3 mois après le début des travaux,
 - 1 équipe d'ouvriers.

Courbe de charge



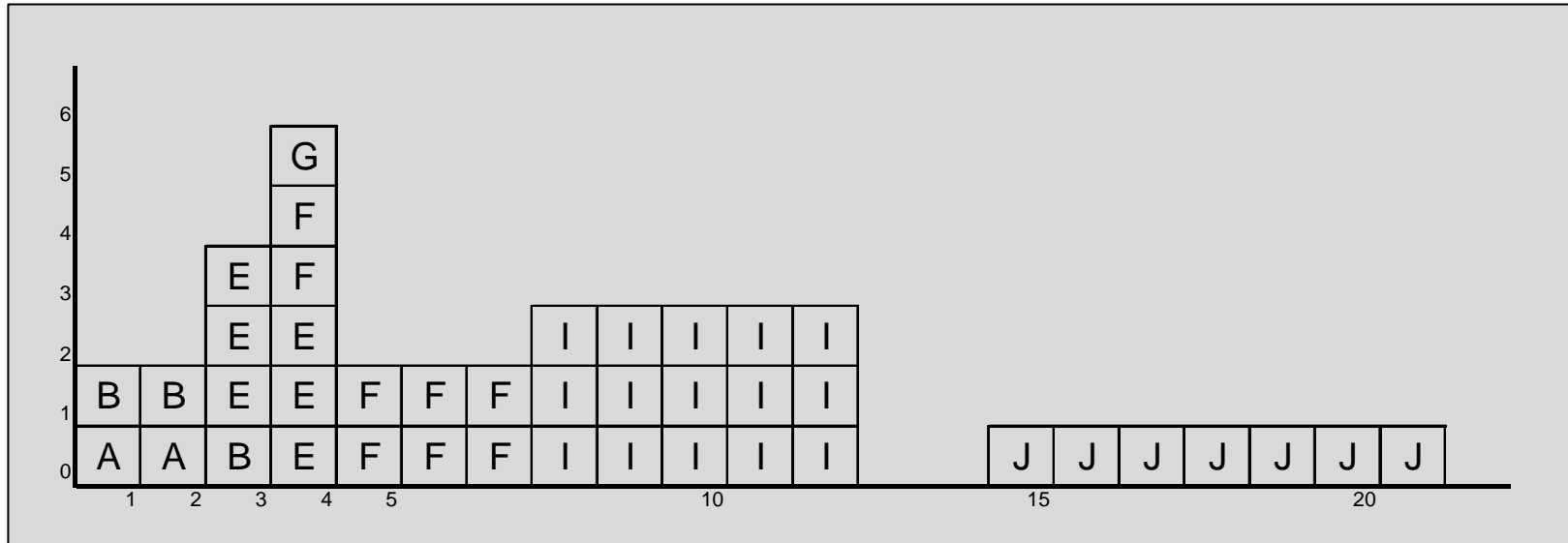
- Tâche I :
 - Durée : 5 mois,
 - Au plus tôt 7 mois après le début des travaux,
 - 3 équipes d'ouvriers.

Courbe de charge



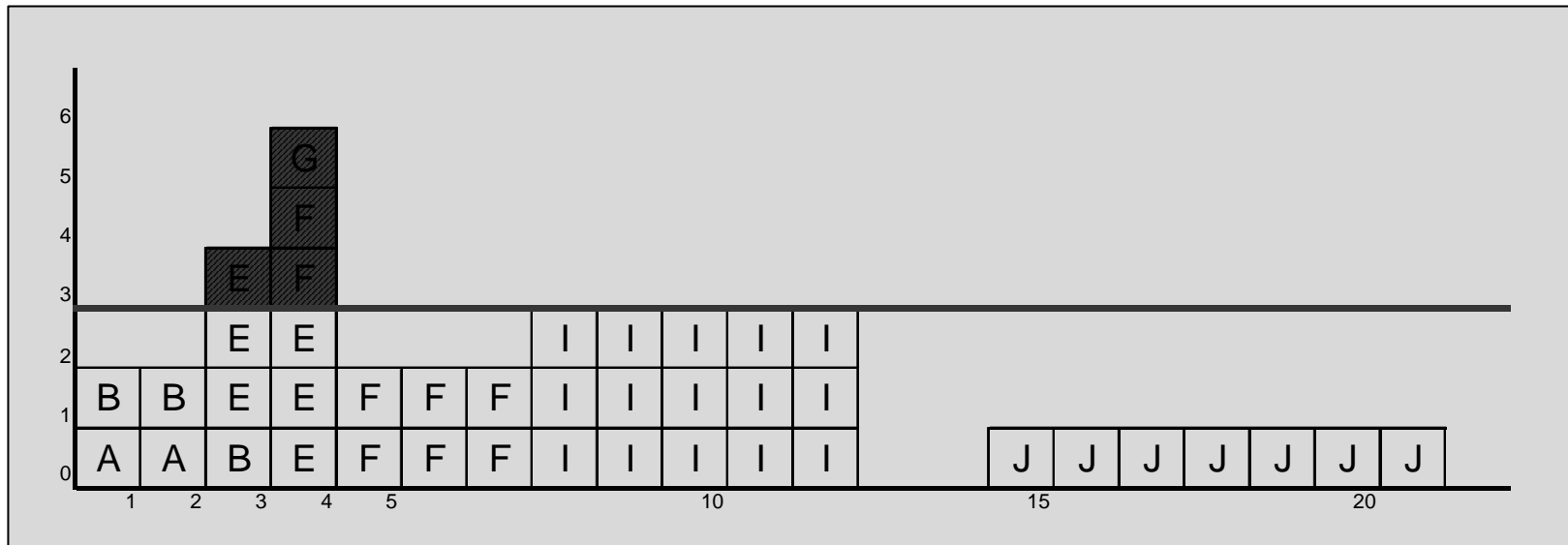
- Tâche J :
 - Durée : 7 mois,
 - Au plus tôt 14 mois après le début des travaux,
 - 1 équipe d'ouvriers.

Courbe de charge



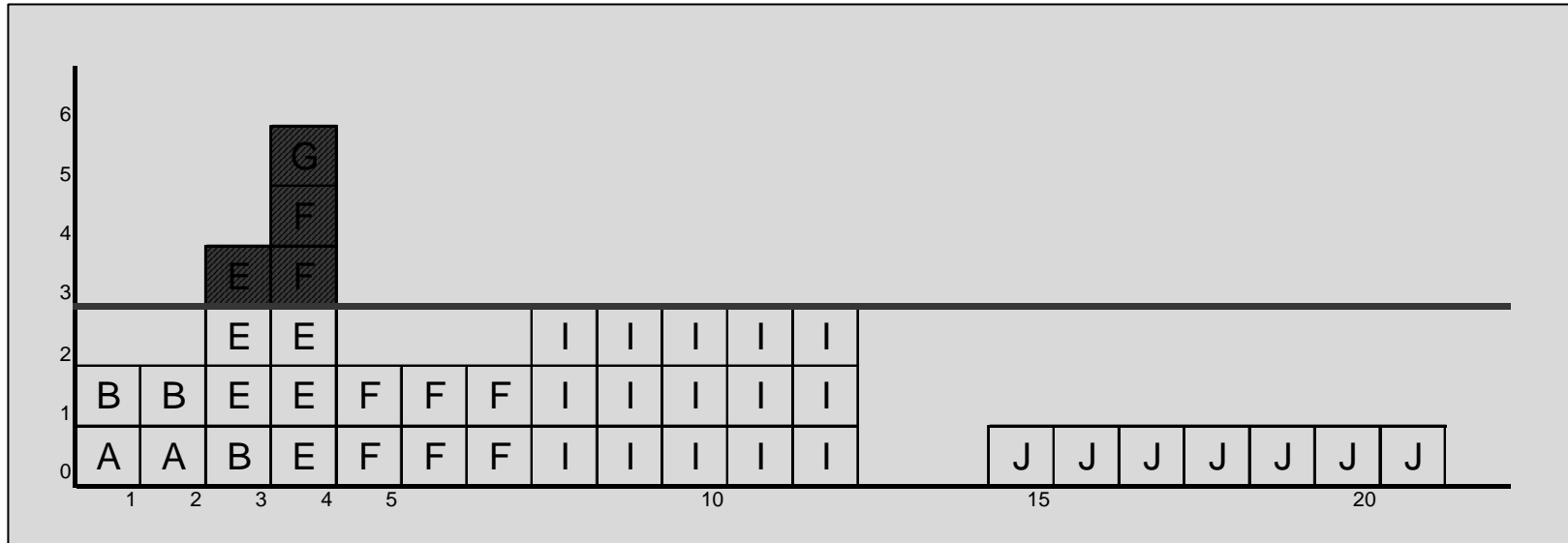
- Courbe de charge peu équilibrée :
 - De 0 à 6 équipes par mois !

Courbe de charge



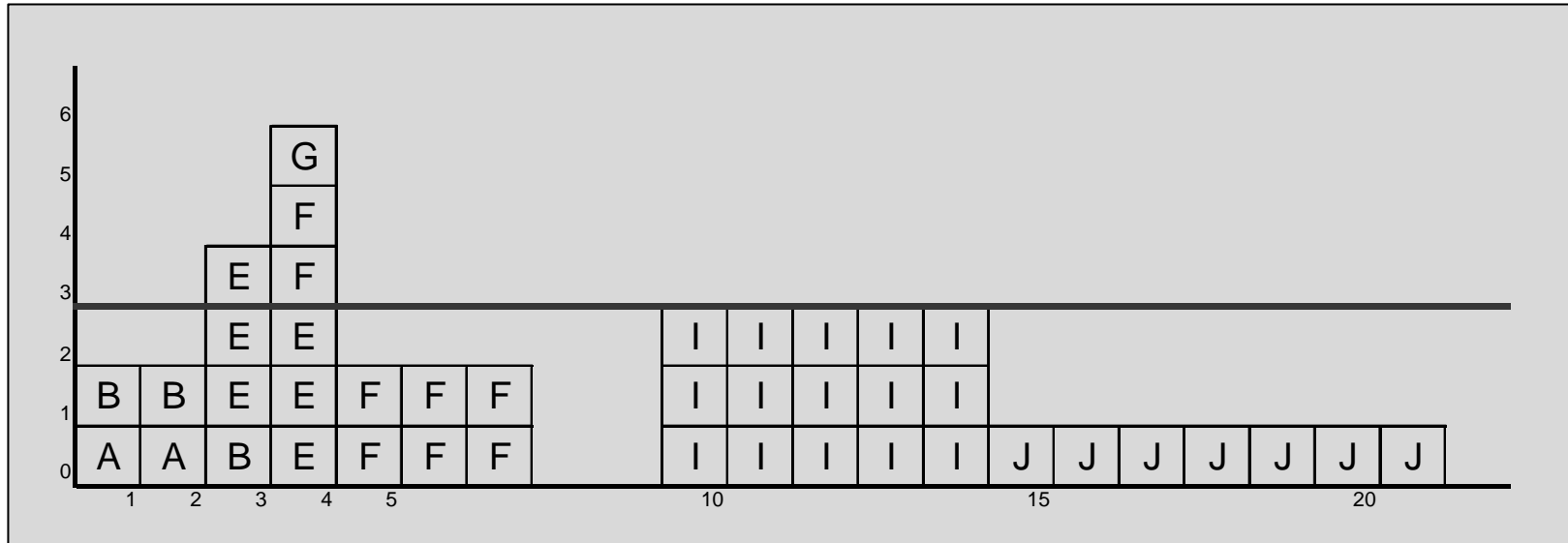
- Contrainte cumulative non respectée :
 - Plus de 3 équipes pendant 2 mois !

Lissage manuel



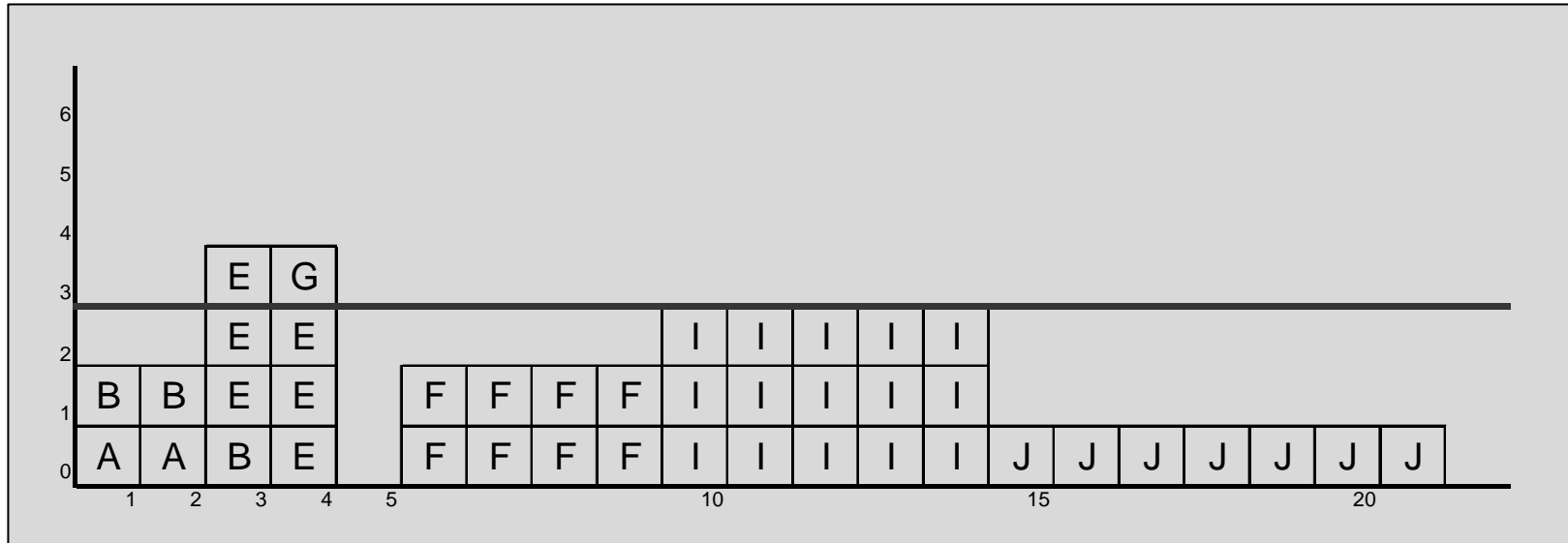
- Reculer la tâche I de 2 mois.

Lissage manuel



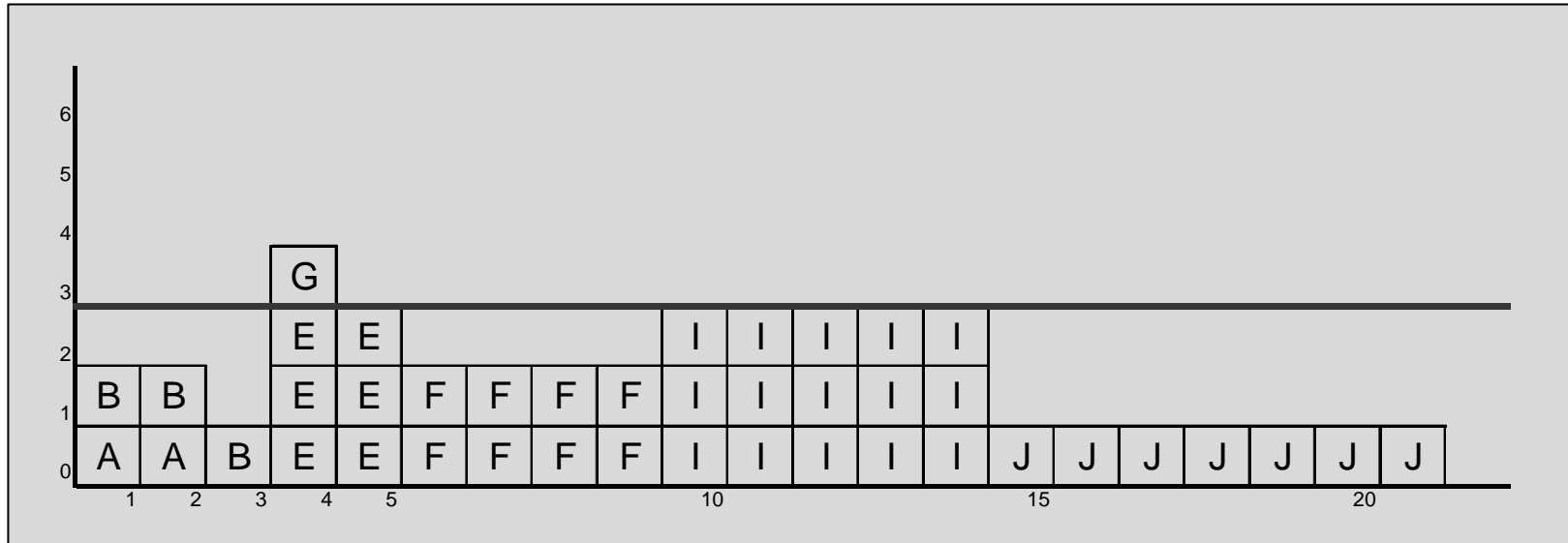
- Reculer la tâche F de 2 mois.

Lissage manuel



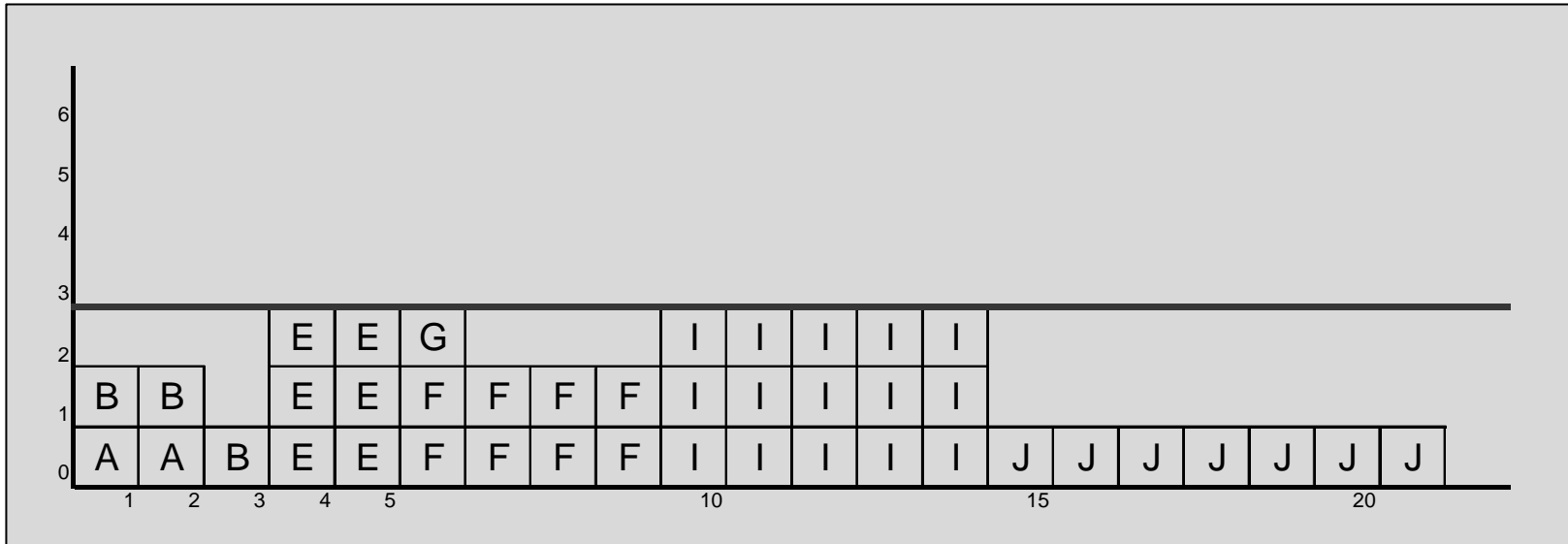
- Reculer la tâche E de 1 mois.

Lissage manuel



- Reculer la tâche G de 2 mois.

Lissage manuel



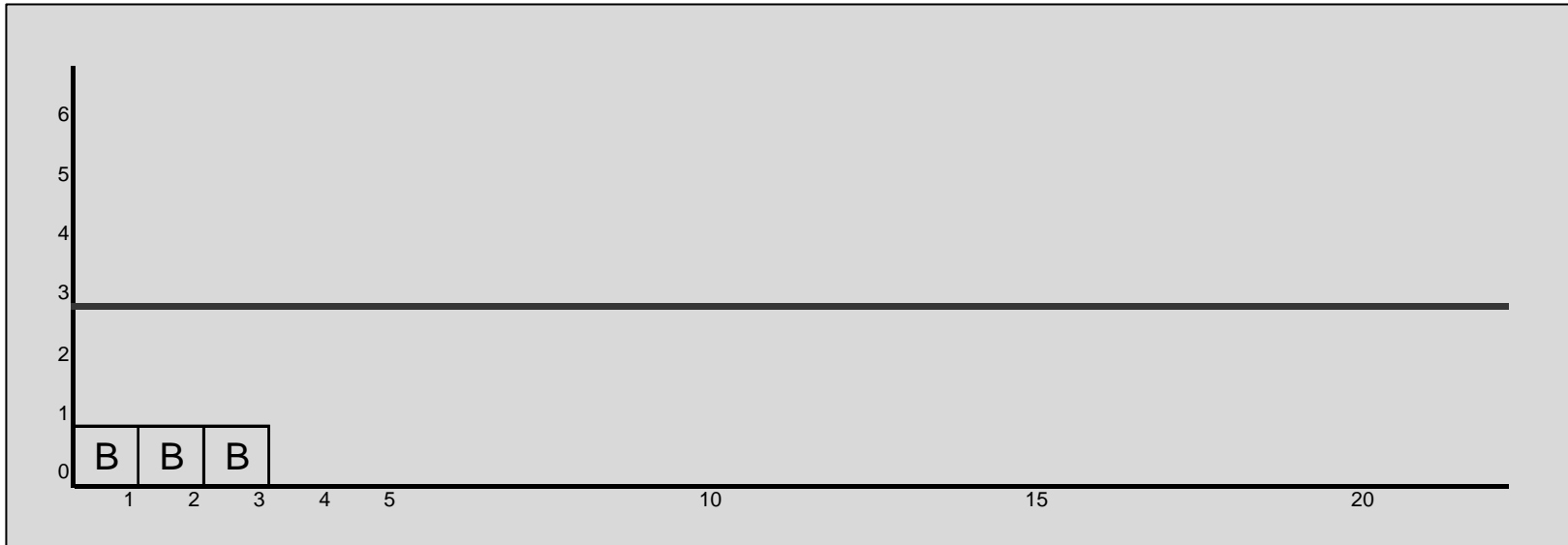
- Contrainte respectée.
- En 21 mois malgré tout (coup de chance) !
- Approche systématique ?

Algorithme MILORD

- Ranger les tâches par ordre croissant de leur date de début au plus tard. Départager les ex-aequo par leur marge libre.
- Placer successivement les tâches au plus tôt, en tenant compte des contraintes.

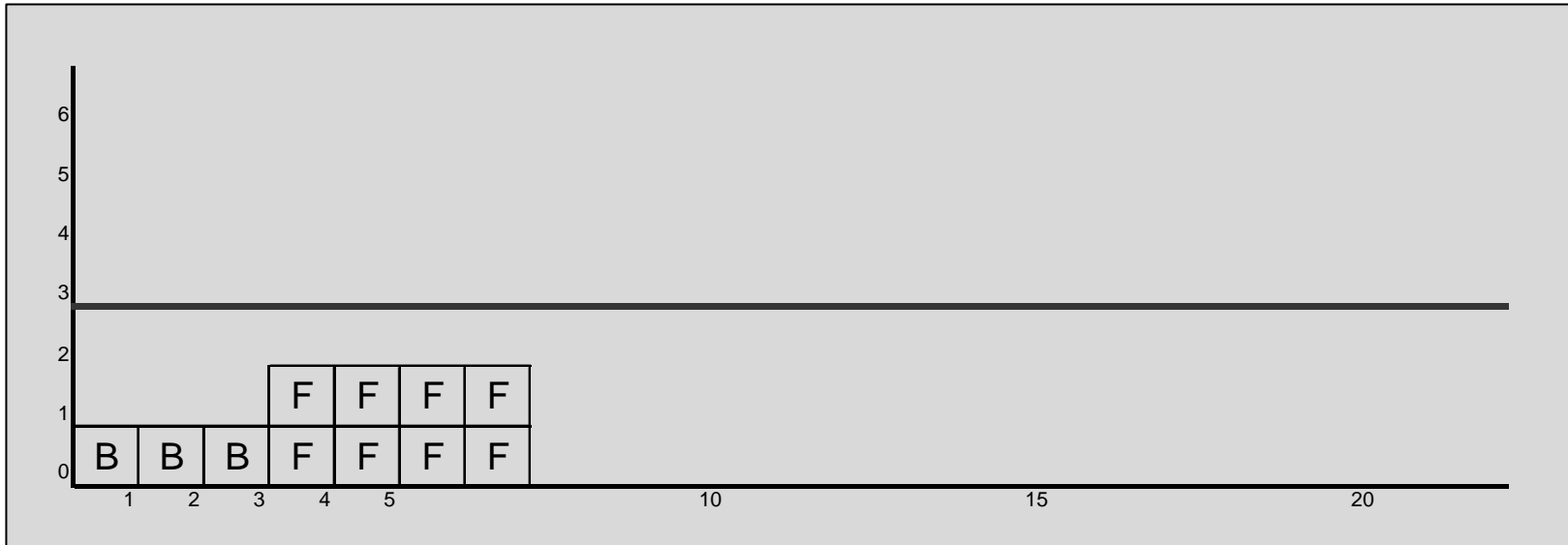
	D	H	B	C	F	A	G	J	E	I
LS	0	6	9	9	12	12	13	14	14	16
ML	0	0	0	9	0	0	10	0	3	9
ES	0	6	0	0	3	0	3	14	2	7

MILORD



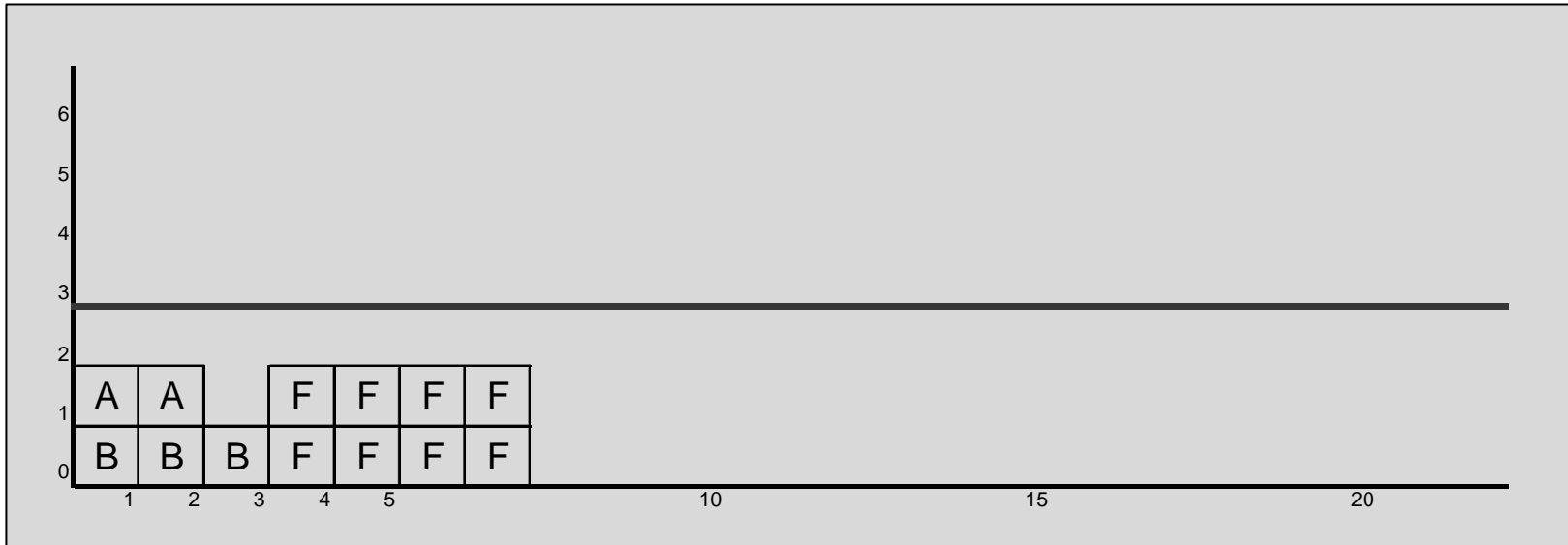
- Placer la tâche F (ES = 3).

MILORD



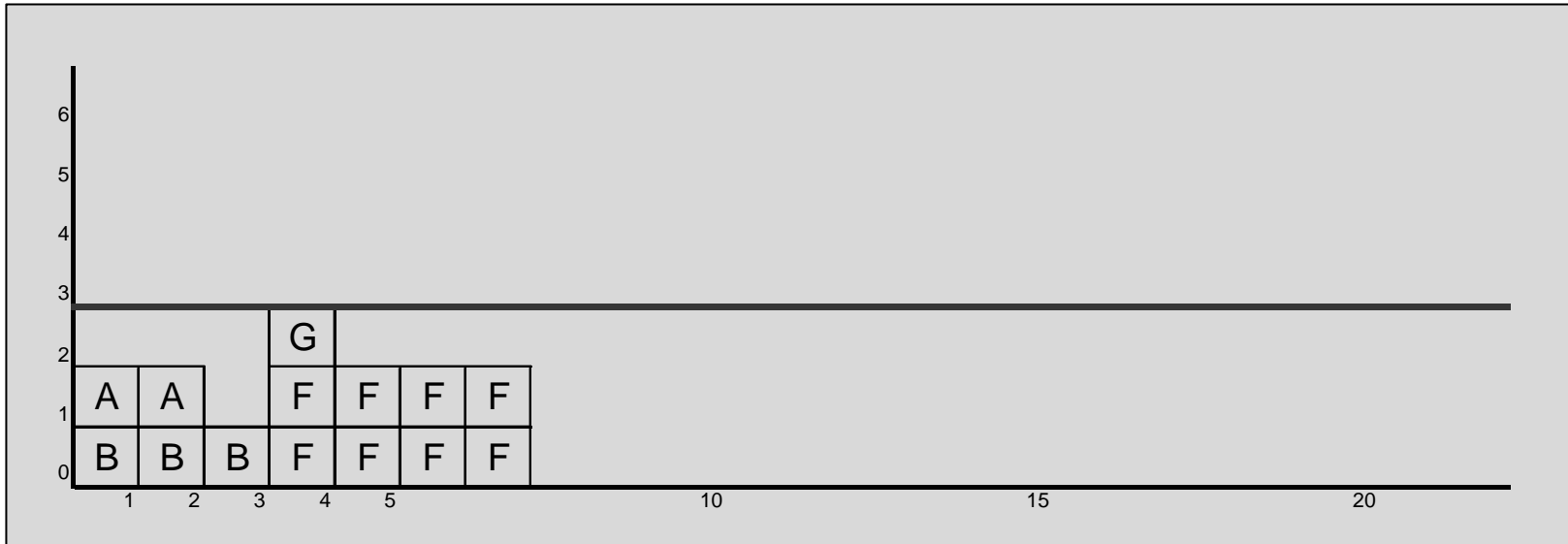
- Placer la tâche A ($ES = 0$).

MILORD



- Placer la tâche G (ES = 3).

MILORD



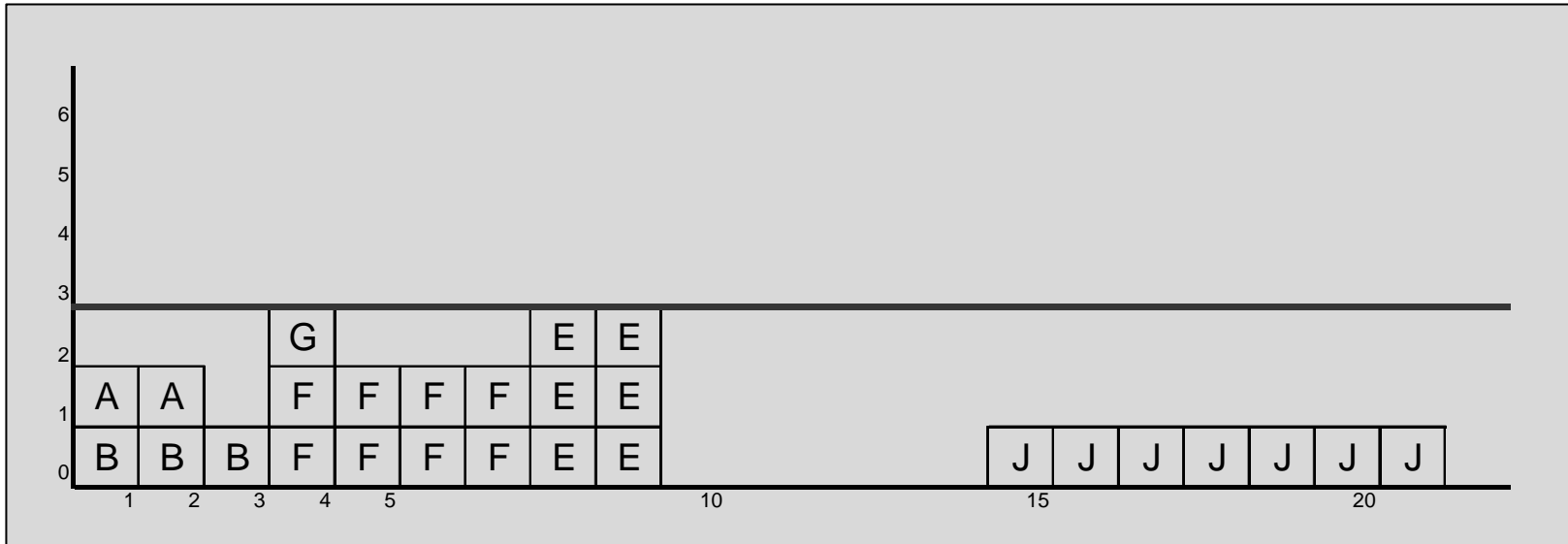
- Placer la tâche J (ES = 14).

MILORD



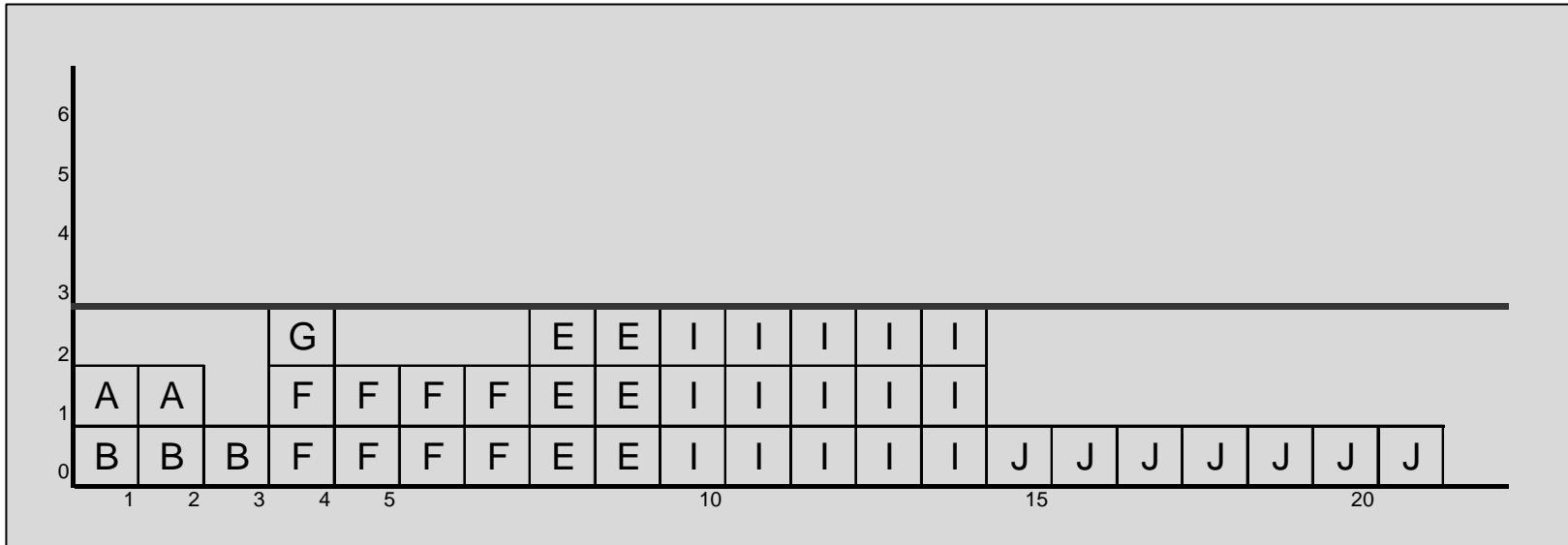
- Placer la tâche E (ES = 2).
- Reculée jusqu'en 7 !

MILORD



- Placer la tâche I (ES = 7).
- Reculée jusqu'en 9 !
- Tout juste...

MILORD



- Fin...

Variantes

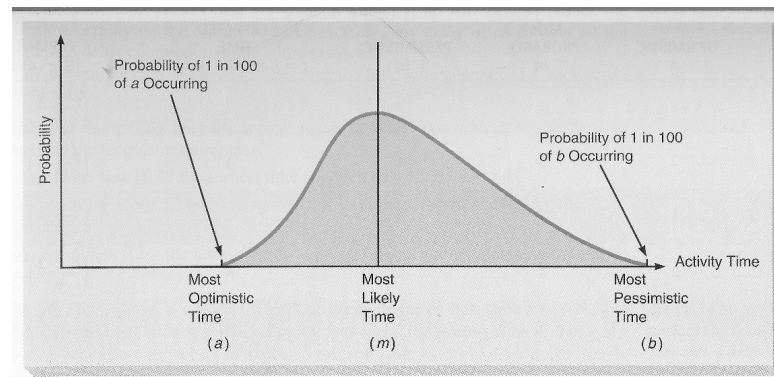
- Méthode des potentiels :
 - Autre mise en graphe : sommets = tâches.
- Méthode PERT :
 - Incertitude sur la durée de réalisation des tâches.
- Prise en compte du coût vs durée de réalisation des tâches.

PERT - Durées des tâches

- Incertitude.
- Distribution de probabilité basée sur 3 estimations (pour chaque tâche):
 - Optimiste (a): durée si tout va pour le mieux, peu vraisemblable (prob $\approx 1\%$).
 - Pessimiste (b): durée si tout va au plus mal, peu vraisemblable (prob $\approx 1\%$).
 - Vraisemblable (m): durée de réalisation la plus vraisemblable.

Distribution Beta

- Hypothèse appropriée dans bcp de cas.



- Durée moyenne :

$$t = \frac{a + 4m + b}{6}$$

- Variance :

$$\text{var} = \left(\frac{b - a}{6} \right)^2$$