Unité d'Enseignement RCP101 : Recherche Opérationnelle et Aide à la Décision

Cours 3 – Ordonnancement

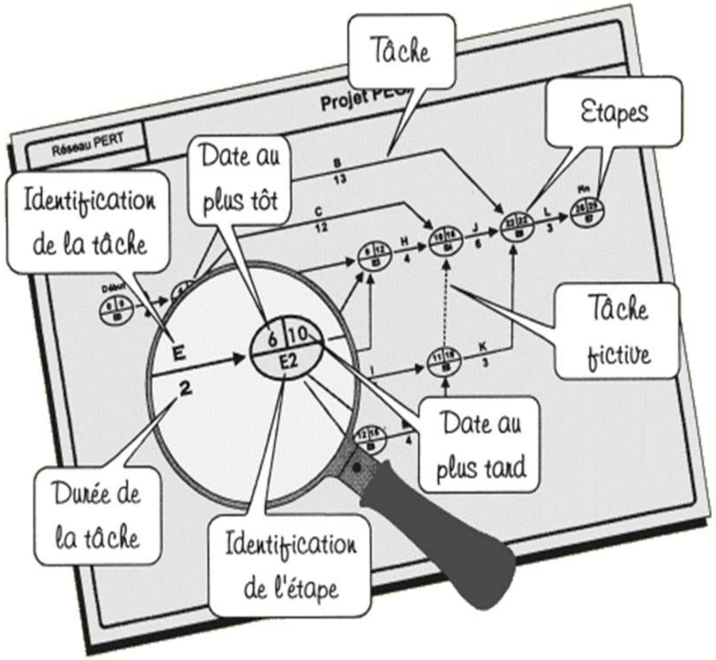
Conservatoire National des Arts et Métiers – E. Soutil

UE RCP101 – Recherche Opérationnelle et Aide à la Décision – Plan du cours

- Partie 1- Eléments de Théorie des Graphes
  - *Généralités, fermeture transitive et connexité*
  - *Chemins de longueur optimale*
- Partie 2 – Ordonnancement
  - *Méthode PERT*
  - *Méthode MPM*
- Partie 3 – Programmation linéaire
  - *Modélisation*
  - *Méthode du simplexe*
  - *Dualité*
- Partie 4 : Processus de Markov et files d'attente
- Partie 5 : Optimisation multicritères

Introduction

- Gestion d'un **projet** : ensemble de tâches concourant à un même but. Ex : projet immobilier
- Mettre en ordre les différentes **tâches**, soumises à des **contraintes**
- **Ordonnancer un projet** : agencer les tâches de sorte que le temps nécessaire à la réalisation de toutes les tâches soit **minimal**
- Il est possible, dans la plupart des cas, de réaliser plusieurs tâches **conjointement** (en parallèle)
- Les méthodes générales datent de 1958 (US Navy et Nasa) :
  - Critical Path Method (**CPM**) et **PERT** (Program Evaluation Research Task)
  - Méthode potentiels-tâches (**MPM**, B. Roy)



le cnam

Données et contraintes du problème

- **Données :**
  - Un projet est composés de n tâches (ou activités) A_i, i = 1 à n,
  - chaque tâche a une durée supposée connue d_i et peut être soumise à un ensemble de contraintes (décrites plus loin)
  - On crée deux tâches fictives : début et fin, de durée nulle, qui correspondent au début et à la fin du projet

Données et contraintes du problème

- **Exemple :** ordonnancement d'atelier

Tâches		Durée	Tâches préalables
n°	Désignation		
a	Enlèvement carcasse	1	-
b	Nettoyage carcasse	2	a
c	Remplacement tôle, visserie défectueuses	1	b
d	Peinture	1	c
e	Nettoyage corps	2	a
f	Tests mécaniques	1	e
g	Tests électriques	2	f
h	Approvisionnement en rechanges mécaniques	2	f, c
i	Approvisionnement en rechanges électriques	1	g
j	Pose des rechanges	2	h, i
k	Remontage	1	d, j
l	Essais	2	k

Données et contraintes du problème

- **Contraintes** :

  Chaque tâche i est caractérisée par :

  - Sa durée d_i (supposée connue au départ)
  - La date t_i (inconnue avant la résolution du problème) à laquelle commencera son exécution

  On distingue habituellement 3 types de contraintes qui caractérisent les tâches :

  - Les contraintes **potentielles** (faciles à prendre en compte)
  - Les contraintes **disjonctives** (qui rendent le problème difficile à résoudre)
  - Les contraintes **cumulatives** (également très dures à prendre en compte)

Contraintes potentielles

- Les contraintes potentielles peuvent facilement être prises en compte. On distingue :
  - Les contraintes de **localisation temporelle** :

    On suppose que l'origine des temps est 0 (date de début du projet).

    Ex. de contrainte de localisation temporelle : $t_i \geq 5$ (la tâche i doit commencer après le 5ème jour)
  - Les contraintes de **précédence**. **Ex** : la tâche i doit être exécutée avant le début de la tâche j :

    $$t_j \geq t_i + d_i$$
  - Les **dates butoirs**. **Ex** : la tâche j commence au plus tard à la fin de la tâche i :

    $$t_j \leq t_i + d_i$$
  - La **synchronisation**. **Ex** : le début de la tâche j doit coïncider avec la fin de la tâche i :

    $$t_j = t_i + d_i$$

Contraintes disjonctives

- Elles rendent le problème **difficile**
- Il s'agit de contraintes pouvant être satisfaites par un choix
- **Ex** : le menuisier est aussi le vitrier, on ne peut entreprendre la pose des vitres (tâche `i`) et celles des placards (tâche `j`) simultanément.
- On a alors :

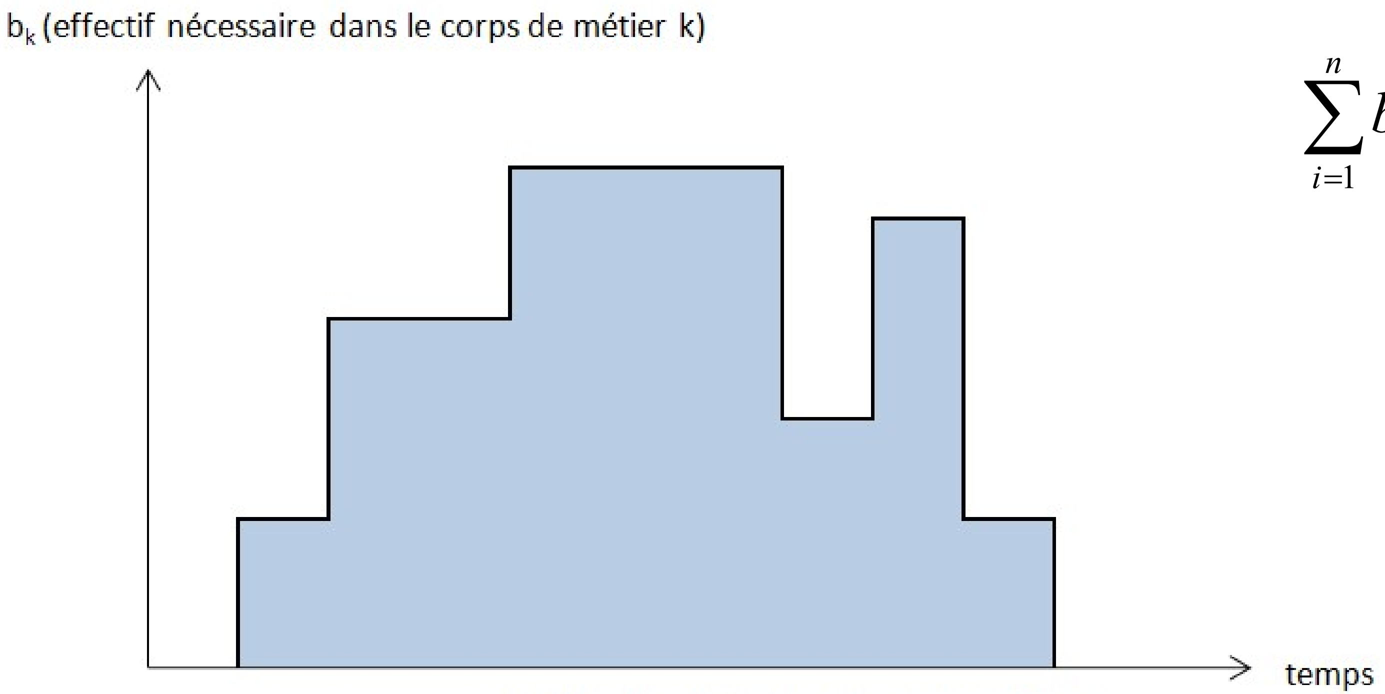
  (tâche `i` avant `j`) ou bien (tâche `j` avant `i`)
- Prise en compte par la PLNE (Programmation linéaire en nombres entiers, cf. plus loin)

Contraintes cumulatives

- Ce type de contraintes est également très du à prendre en compte : elles sont généralement ignorées, au risque de ne pas ordonnancer au moindre coût
- Prise en compte de la **limitation des ressources** (crédits, équipent, main d'œuvre)
- On associe à la tâche `i` une fonction b_{ik} qui mesure à chaque instant les besoins en main d'œuvre pour le corps de métier `k`.
- Si on cumule les valeurs b_{ik} pour l'ensemble des n tâches, on obtient une **courbe de charge** du corps de métier k.

$$\sum_{i=1}^{n} b_{ik} = b_k$$

Contraintes cumulatives



$$\sum_{i=1}^{n} b_{ik} = b_k$$

Courbe de charge du corps de métier k

- On doit alors tenir compte de l'évolution au cours du temps des besoins globaux en main d'œuvre dans le corps de métier k, dont on peut vouloir limiter les variations

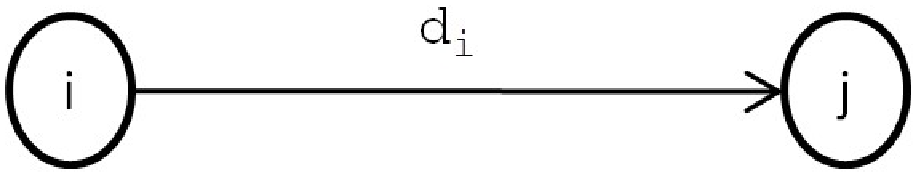
Contraintes cumulatives

- Pour prendre en compte ces contrainte, on effectue une résolution dite **heuristique** (ou approchée)
- **Exemple** de méthode heuristique utilisée : la **méthode sérielle**

  On met en ordre les tâches en tenant compte des contraintes de charge et des contraintes de précédence, en donnant la priorité aux tâches vérifiant un critère donné (cf ED).

La méthode potentiel-tâche (MPM)

- Due à B. Roy (Méthode Potentiel Métra, du nom de la société de conseil qui étudiait la construction du paquebot France)
- La méthode s'applique lorsqu'il n'y a que des contraintes potentielles
- On construit un graphe :
  - **Un sommet = une tâche** (y compris les tâches fictives de du début et de la fin du projet)
  - **Un arc (i, j)** représente une **contrainte de précédence** et est valué par la durée d_i de la tâche i :



- **Contraintes implicites** :
  - La tâche de début précède toutes les tâches sans prédecesseur
  - Toute tâche sans successeur précède la tâche de fin

La méthode potentiel-tâche (MPM)

- **But recherché** : on désire calculer :

  - La durée minimale **D** d'exécution du projet entier

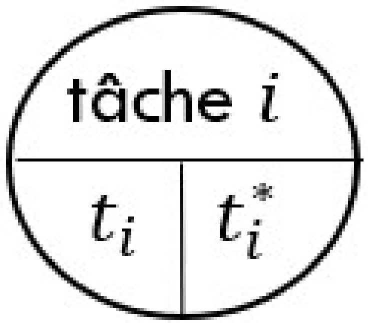
  - La date t_i à laquelle chaque tâche i peut commencer **au plus tôt** (compte tenu des contraintes de précédence)

  - La date t_i^* à laquelle chaque tâche i doit commencer **au plus tard** si on ne veut pas retarder la fin globale du projet (la différence entre t_i et t_i^* est appelée marge totale)

  - Les **tâches critiques** : celles sur lesquelles tout retard pris décale d'autant la fin globale du projet.

La méthode potentiel-tâche (MPM) – **Première étape : tracé du graphe**

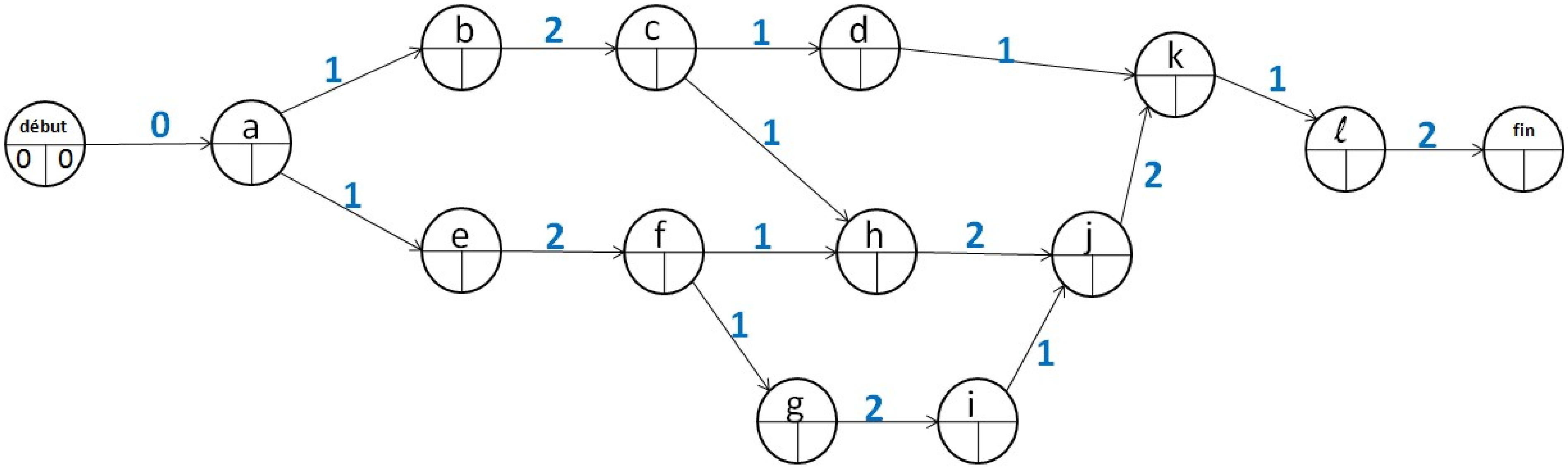
- Un sommet comprend trois informations :
  - le **nom** de la tâche i
  - la date de début au plus tôt t_i
  - et la date de début au plus tard t_i^* (si la tâche débute après cette date, le projet sera retardé)



le cnam

La méthode potentiel-tâche (MPM) – **Première étape : tracé du graphe** – L'exemple

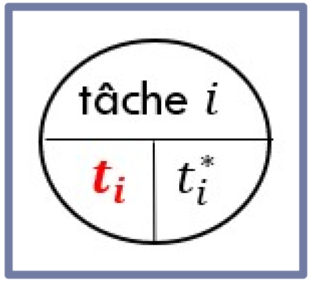
On reprend les données de l'exemple « ordonnancement d'atelier » (page 5) :



La méthode potentiel-tâche (MPM) – **Deuxième étape : détermination des dates au plus tôt t_i**

- On part de la tâche de début qui a 0 comme date au plus tôt
- Si la tâche j précède la tâche i, alors : $t_i \geq t_j + d_j$
- On a donc :

$$\boxed{t_i = \max_{j \in \Gamma^-(i)} (t_j + d_j)}$$

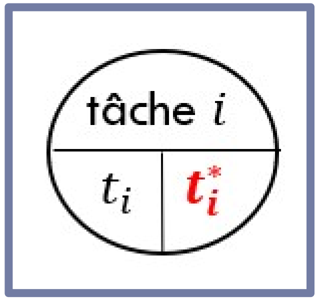


- De proche en proche, le graphe étant sans circuit, on trouve ainsi la date au plus tôt de la fin du projet, qui devient également la date au plus tard de la fin du projet. Il s'agit de la durée **D** minimale du projet.
- **Remarque** : calculer les dates de début au plus tôt des tâches revient à appliquer l'algorithme de Ford (version plus long chemin) sur un ordre topologique des sommets (i.e. l'algorithme de Bellman), les dates t_i ainsi trouvées correspondent à la longueur d'un plus long chemin du sommet début au sommet i.

La méthode potentiel-tâche (MPM) – **Troisième étape : détermination des dates au plus tard t_i^***

- On part cette fois de la tâche de fin. Sa date au plus tard est connue : c'est **D** (sinon le projet prend du retard) ;
- Si la tâche i précède la tâche j, alors : $t_i^* + d_i \leq t_j^*$
- On a donc :

$$\boxed{t_i^* = \min_{j \in \Gamma^+(i)} (t_j^* - d_i)}$$



- Ainsi, la date au plus tard de toute tâche se déduit de celles de ses successeurs, de proche en proche également.

- Remarque : calculer les dates de début au plus tard t_i^* revient à calculer la distance (longueur d'un plus court chemin) du sommet fin au sommet i sur le graphe inverse, obtenu à partir de G en inversant le sens de tous les arcs

La méthode potentiel-tâche (MPM) – **Quatrième étape : détermination des tâches critiques**

- Le plus long chemin du graphe est appelé **chemin critique** et les tâches qui le composent **tâches critiques**

- En effet, le long de ce chemin, toutes les tâches doivent être exécutées : la durée du projet est donc au moins égale à la longueur de ce chemin

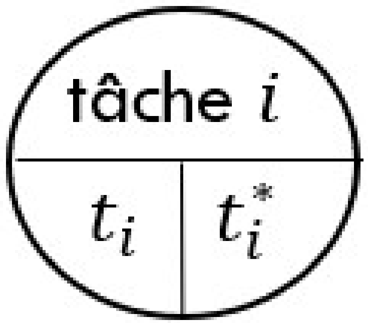
- On peut montrer que la durée du projet est exactement la longueur de ce chemin, toutes les autres tâches pouvant être exécutées « en parallèle »

- Si le graphe est sans circuit : algorithme de Bellman (qui correspond à l'application de l'algorithme de Ford sur un ordre topologique des sommets)

La méthode potentiel-tâche (MPM) – **Cinquième étape : calcul des marges**

- La **marge totale M_i** d'une tâche i est le délai dont on peut retarder le début de cette tâche sans affecter la date globale D d'achèvement du projet :

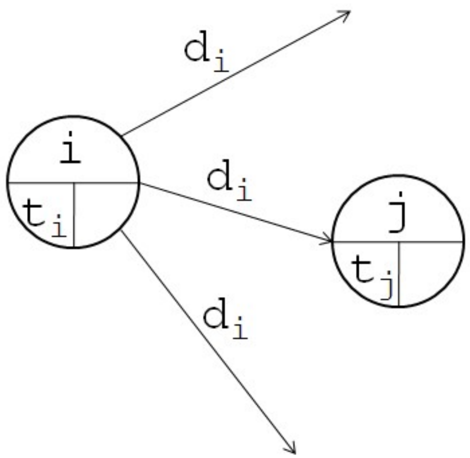
$$\boxed{M_i = t_i^* - t_i}$$



La méthode potentiel-tâche (MPM) – **Cinquième étape : calcul des marges**

- La **marge libre m_i** d'une tâche i est le délai dont on peut retarder le début de cette tâche sans affecter les dates de début au plus tôt des tâches postérieures (si on veut qu'aucun corps de métiers n'empiète sur la marge d'un corps de métier intervenant après lui, pour ne pas le « presser » en quelques sortes, i.e. qu'il puisse lui-même commencer au plus tôt) :

$$\boxed{m_i = \min_{j \in \Gamma^+(i)} (t_j - t_i - d_i)}$$

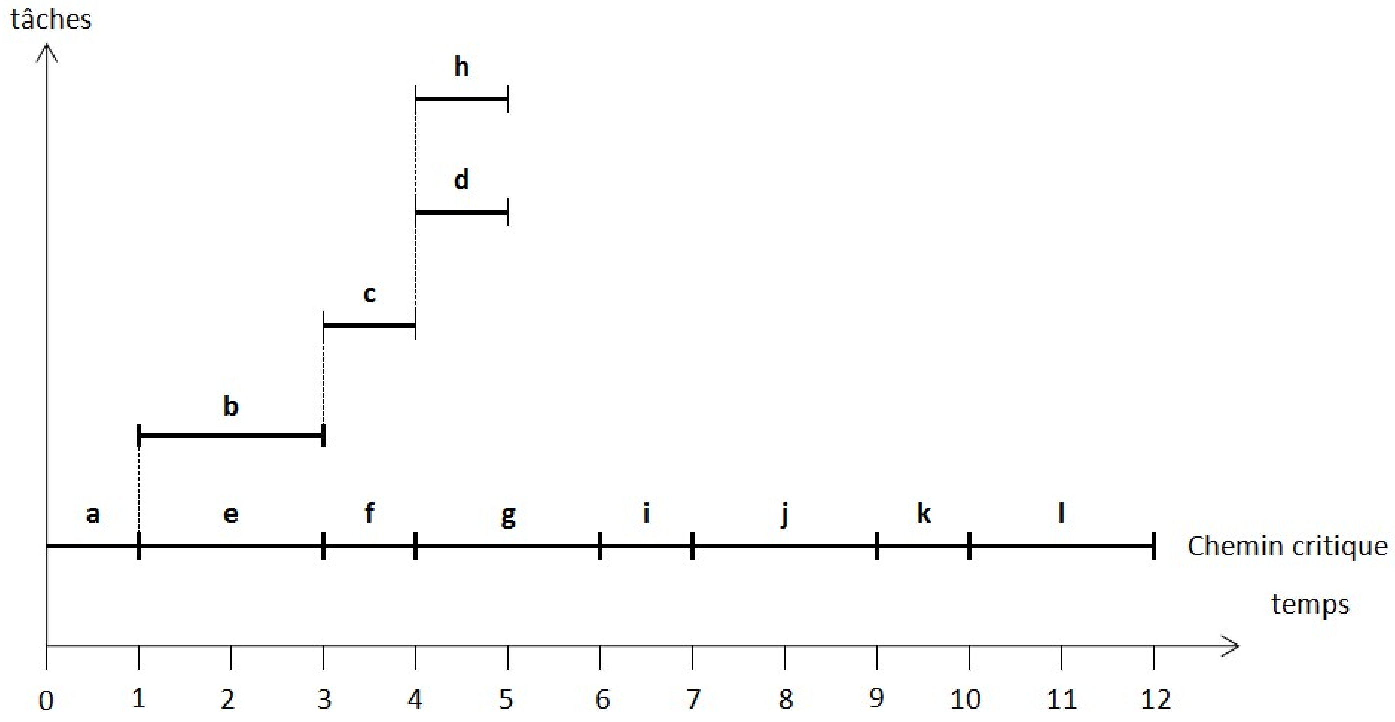


- En effet, on doit avoir :
  $\forall j \in \Gamma^+(i), t_j \geq t_i + m_i + d_i$
  c'est-à-dire : $\forall j \in \Gamma^+(i), m_i \leq t_j - t_i - d_i$

le cnam

La méthode potentiel-tâche (MPM) – **Diagramme de Gantt**

- À partir des dates au plus tôt calculée pour chaque tâche, on peut tracer **le diagramme de Gantt** qui donne une autre vision du déroulement du projet :



La méthode PERT

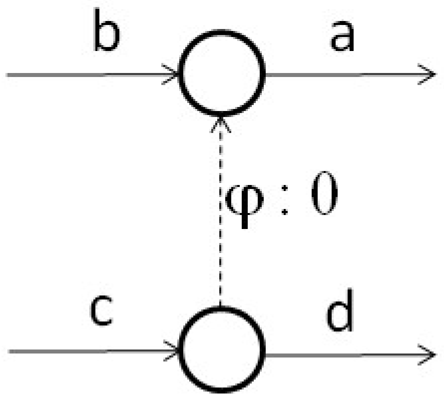
- **Une tâche = un arc**, valué par la durée d_i de la tâche i
- Les **sommets** représentent les « **étapes** » (appelées aussi événements) du projet
- Une étape : instant à partir duquel une ou plusieurs tâches peuvent commencer
- PERT ou MPM : schémas sensiblement différents
- Méthode des potentiels (MPM) plus pratique pour plusieurs raisons :
  - Elle est mieux adaptée à des modifications incessantes
  - Difficulté de la méthode PERT : nécessité d'introduire des tâches fictives de durée 0 pour tenir compte de toutes les contraintes de précédence

La méthode PERT

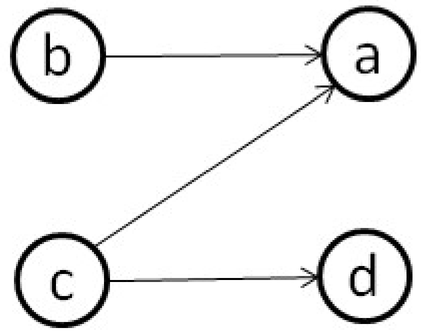
- **Exemple :**

Tâches	Tâches préalables
a	b, c
d	c

- Avec la méthode PERT, on est obligé pour représenter les précédence d'introduire une **tâche fictive φ** de durée 0, ce qui n'est pas nécessaire avec MPM :



PERT



MPM

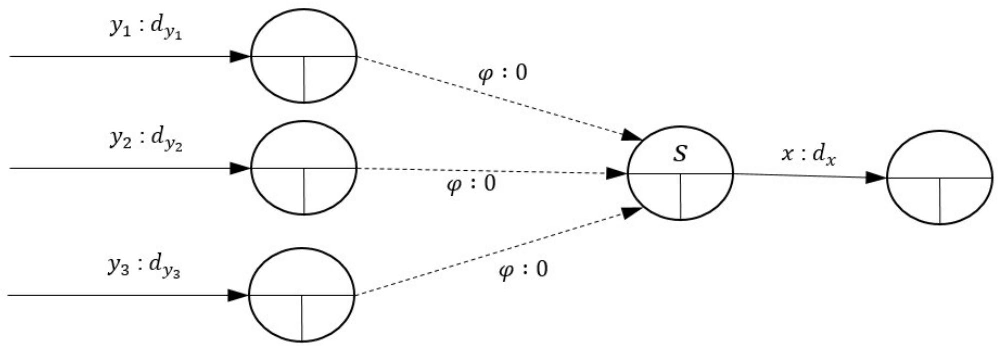
La méthode PERT

- Dans les deux méthodes, les dates au plus tôt et au plus tard se calculent de la même façon, mais en pratique, on a en général besoin des dates de début au plus tôt de chaque tâche, que l'on obtient avec MPM, plutôt que de chaque étape : la méthode MPM reste là encore plus lisible

le cnam

La méthode PERT – tracé du graphe

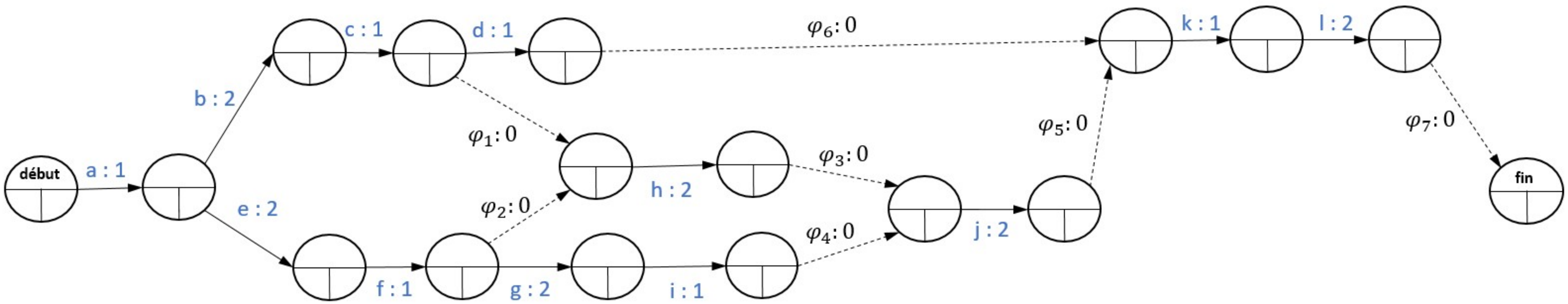
- On place le sommet début;
- Les tâches sans tâche préalable partent du sommet début;
- Une tâche x ayant exactement une tâche préalable y se trace en faisant partir l'arc x depuis le sommet correspondant à la fin de l'arc y;
- Si une tâche x a plusieurs tâches préalables $y_1, y_2, y_3, \ldots$, on crée un nouveau sommet s à partir duquel va partir la tâche x et on relie le sommet fin de chacune des tâches $y_1, y_2, y_3, \ldots$ au sommet s par une tâche fictive φ de durée nulle :



- Puis on simplifie les tâches fictives inutiles selon le procédé décrit plus loin

La méthode PERT – Exemple de graphe PERT

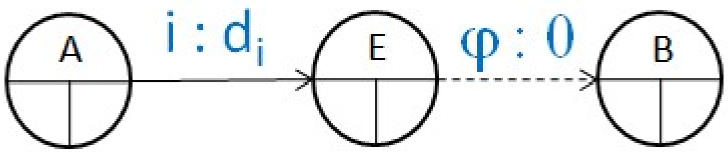
- On reprend l'exemple de l'ordonnancement d'atelier.
- En appliquant la méthode de tracé (systématique) du graphe PERT on obtient :



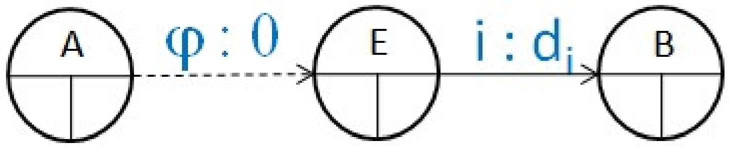
- Sur ce graphe, nous allons pouvoir simplifier plusieurs tâches fictives.

La méthode PERT – simplification du graphe

- **Remarque** : après avoir introduit des tâches fictives pour représenter les contraintes de précédence, il est parfois possible de simplifier le graphe. Si l'on rencontre l'une des deux configurations suivantes, dans lesquelles le sommet E **n'a qu'un unique successeur et un unique prédécesseur** :



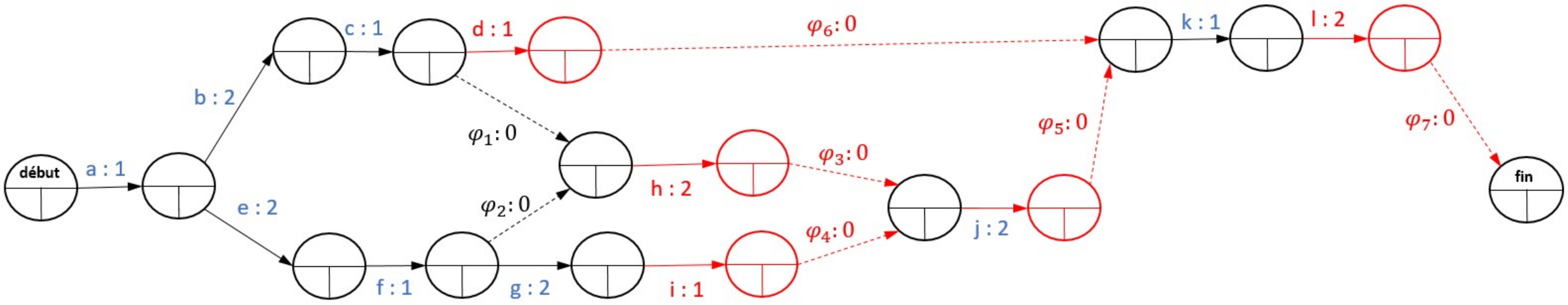
ou



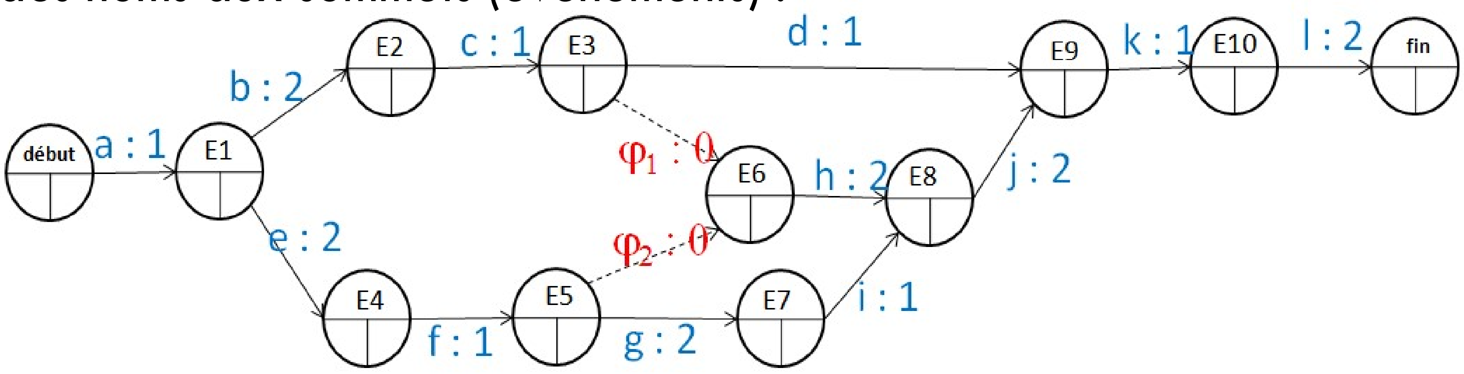
… il est alors possible de **supprimer la tâche fictive** et on aboutit à :

La méthode PERT – Exemple de graphe PERT (suite)

- Sur le graphe précédent apparaissent plusieurs simplifications possibles (indiquées en rouge), permettant de supprimer les tâches fictives inutiles $\varphi_3, \varphi_4, \varphi_5, \varphi_6$ et φ_7 :

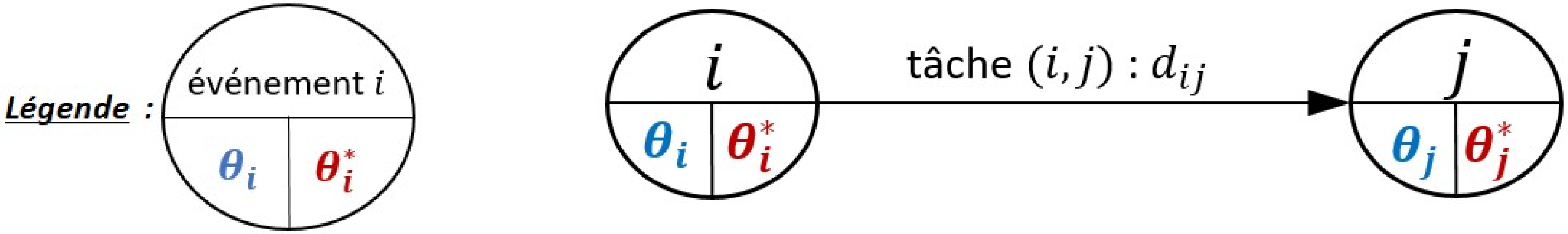


- On aboutit alors au graphe PERT final, dans lequel on a donné arbitrairement des noms aux sommets (événements) :



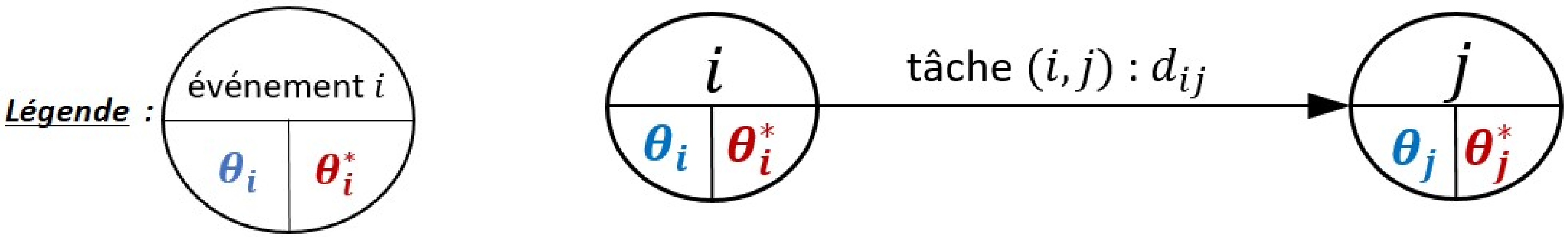
le cnam

La méthode PERT – Calcul des dates au plus tôt, au plus tard, marges libres et totales



- θ_i = date au plus tôt de l'événement i : $\theta_i = \max\limits_{j \in \Gamma^-(i)} (\theta_j + d_{ji})$, avec $\theta_{deb} = 0$
- θ_i^* = date au plus tard de l'événement i : $\theta_i^* = \min\limits_{j \in \Gamma^+(i)} (\theta_j^* - d_{ij})$, avec $\theta_{fin}^* = \theta_{fin}$
- m_{ij}= **marge libre** de la tâche (i, j). C'est le délai maximal dont on peut retarder le début de la tâche (i, j) sans affecter le début au plus tôt des tâches qui suivent : $\theta_i + m_{ij} + d_{ij} = \theta_j \Rightarrow \boldsymbol{m_{ij} = \theta_j - \theta_i - d_{ij}}$
- M_{ij}= **marge totale** de la tâche (i, j). C'est le délai maximal dont on peut retarder le début de la tâche (i, j) sans affecter la fin globale du projet : $\theta_i + M_{ij} + d_{ij} = \theta_j^* \Rightarrow \boldsymbol{M_{ij} = \theta_j^* - \theta_i - d_{ij}}$

La méthode PERT – Calcul des dates de début au plus tôt et au plus tard des tâches



- **la date de début au plus tôt $\boldsymbol{t_{ij}}$ de la tâche $\boldsymbol{(i,j)}$** est θ_i (l'événement i correspond au début de la tâche (i,j)) $\Rightarrow \boldsymbol{t_{ij} = \theta_i}$
- **la date de début au plus tard de $\boldsymbol{t_{ij}^*}$ de la tâche $\boldsymbol{(i,j)}$** se déduit de la marge totale : $t_{ij}^* = t_{ij} + M_{ij} = \theta_i + \theta_j^* - \theta_i - d_{ij} \Rightarrow \boldsymbol{t_{ij}^* = \theta_j^* - d_{ij}}$

Prise en compte des contraintes disjonctives

- Ecriture du problème sous forme d'un programme linéaire (minimiser t_{fin} sous les contraintes de précédence)

$$\begin{cases} \text{minimiser } t_{\text{fin}} \\ \text{sous les contraintes} \left| \begin{array}{ll} t_{\text{fin}} \geq t_i & i = 1 \text{ à } n \\ t_i \geq 0 & i = 1 \text{ à } n \\ t_j \geq t_i + d_i & \text{si la tâche } i \text{ précède la tâche } j \end{array} \right. \end{cases}$$

- Une contrainte du type…

$$t_j \geq t_i + d_i \quad \text{ou bien} \quad t_i \geq t_j + d_j$$

… est alors prise en compte à l'aide d'une variable 0-1 x_{ij} :

$$\begin{cases} x_{ij} t_j \geq x_{ij}(t_i + d_i) \\ (1 - x_{ij}) t_i \geq (1 - x_{ij})(t_j + d_j) \end{cases}$$

- Le problème est NP-difficile