Méthode des Potentiels Metra
MPM

2024/2025

Exemple de données

Un étude préalable a permis de décomposer un projet en plusieurs tâches soumises à des contraintes d'antériorité. Le résultat de cette étude est synthétisé dans un tableau de la forme suivante :

tâches	a	b	c	d	e	f	g	h	i
durée	3	8	2	7	5	2	4	3	4
tâches précédentes	$-$	$-$	a, b	a, b	b	b	c, d	d	e, f

Le projet est ici décomposé en 9 tâches (désignées par les lettres de a à i). Pour chaque tâche, ce tableau donne sa durée d'exécution ainsi que ses contraintes d'antériorité.

Par exemple, l'exécution de la tâche c dure 2 (jours, semaines ou autres) et ne peut débuter qu'une fois les tâches a et b terminées.

La méthode et les objectifs

La Méthode des Potentiels Metra (MPM) réalise un ordonnancement ou planification des tâches composant un projet.

Elle établit un calendrier d'exécution des tâches qui respecte les contraintes d'antériorité et qui minimise la durée nécessaire pour mener à bien le projet.

Le calendrier contient différents éléments permettant au chef de projet de piloter et de contrôler l'exécution du projet. Il fixe par exemple la date de début d'exécution de chaque date. Il permet aussi au chef de projet de réagir efficacement si des événements imprévus retardent l'exécution d'une ou plusieurs tâches.

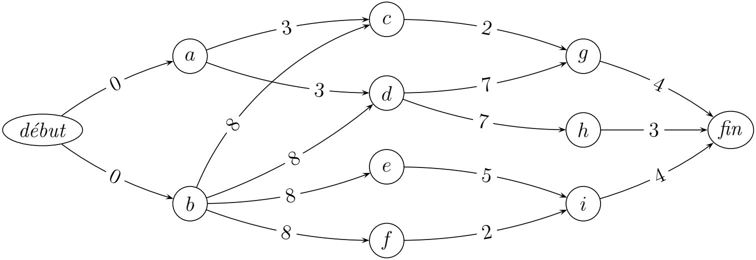
Cette méthode a été mise au point par la société française (*Sema-Metra*) en 1958 afin de résoudre les problèmes posés par la construction du paquebot *France* et par la construction de centrales électriques de la société *EDF*. Cette méthode, simple et efficace, a ensuite été appliquée dans de nombreux projets industriels.

À partir des données, on construit un graphe $G = (S, A, p)$ appelé **graphe potentiel tâches** :

- Les sommets représentent les tâches.
- Les arcs représentent les contraintes d'antériorité. Un arc (t_1, t_2) indique que la tâche t_1 précède la tâche t_2 et son poids est égal à la durée de la tâche t_1.
- On ajoute un sommet noté ***début*** symbolisant le début des travaux. Puis pour tout sommet t sans prédécesseur, on ajoute un arc $(\textit{début}, t)$ de poids 0.
- On ajoute un sommet noté ***fin*** symbolisant la fin des travaux. Puis pour tout sommet t sans successeur, on ajoute un arc $(t, \textit{fin})$ dont le poids est égal à la durée de t.

Le graphe potentiel tâches est un graphe sans circuit (DAG) qui admet une unique source (*début*) **et un unique puits** (*fin*).

tâches	a	b	c	d	e	f	g	h	i
durée	3	8	2	7	5	2	4	3	4
tâches précédentes	–	–	a, b	a, b	b	b	c, d	d	e, f



$(\textit{début}, a, b, c, d, e, f, g, h, i, \textit{fin})$ est un ordre topologique pour ce graphe et c'est aussi un ordonnancement possible des tâches : si les tâches sont exécutées (une à une) dans cet ordre, les contraintes sont respectées et la durée du projet est de 38. Que devient cette durée si les tâches a et b débutent à la date 0 ?

La date de début du projet est fixée à 0. Il s'agit ensuite de déterminer, pour chaque tâche t, la date **minimum** à laquelle l'exécution de t peut débuter.

Cette date est appelée date de début au plus tôt de la tâche t et on la désigne par $tôt(t)$. On a : $tôt(début) = 0$.

Soient (x, t) un arc et C le chemin allant de *début* à t et se terminant par l'arc (x, t). La tâche t ne peut débuter qu'une fois toutes les tâches se trouvant sur le chemin C sont terminées, ce qui prend un temps égal au poids de C. Et puisque plusieurs tâches peuvent être antérieures à t, il vient que :

$tôt(t)$ est égale au poids d'un plus long chemin allant du sommet *début* au sommet t

D'où le calcul des dates de début au plus tôt des taches prises dans un ordre topologique pour la graphe potentiel tâches :

$$tôt(début) = 0 \quad \text{et} \quad tôt(t) = \max_{x \in V^-(t)} \{\, tôt(x) + durée(x) \,\}$$

La **durée minimale du projet**, notée $D_{\min}$, est donc égale à $tôt(fin)$.

L'algorithme de Bellman **qui détermine une arborescence de chemins de poids maximum de racine *début*** est le suivant :

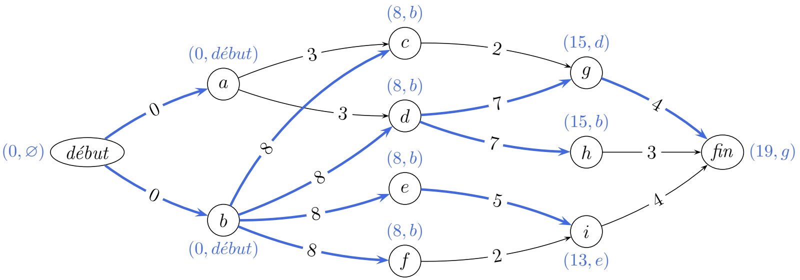
- **Initialisation des potentiels :** $\Pi[\textit{début}] = 0$ et $\Pi[t] = -1$ pour tout sommet $t \neq \textit{début}$ (puisque les durées sont positives).

- **Itérations :** pour chaque sommet t du graphe pris dans **un ordre topologique**, calculer $\mathbf{\Pi[t] = \max_{x \in V^-(t)}\{\ \Pi(x) + p(x,t)\ \}}$. Cela qui revient à **relâcher tous les arcs de but t**, le relâchement d'un arc (x,t) s'écrivant ici :

  si $\Pi[x] + p(x,t) > \Pi[t]$ **alors**
  $\quad\Pi[t] \leftarrow \Pi[x] + p(x,t)\,;\ \mathcal{P}[t] \leftarrow x$
  fin si

- **Résultats :** pour tout sommet t du graphe, $\textit{tôt}(t) = \Pi[t]$.

On utilise l'ordre topologique $(\textit{début}, a, b, c, d, e, f, g, h, i, \textit{fin})$:



t	*début*	a	b	c	d	e	f	g	h	i	*fin*
$\boldsymbol{tôt(t) = \Pi(t)}$	**0**	**0**	**0**	**8**	**8**	**8**	**8**	**15**	**15**	**13**	**19**
père : $\mathcal{P}(t)$	$\varnothing$	*début*	*début*	b	b	b	b	d	d	e	g

Dates de début au plus tard

Le calcul des dates de début au plus tôt a établi la durée minimale du projet à $D_{\min} = \textit{tôt}(\textit{fin})$. Pour chaque tâche t, on s'intéresse maintenant à la date **maximum** à laquelle l'exécution de t peut débuter sans que cela ne diffère $\textit{tôt}(\textit{fin})$ (ou augmente la durée prévue $D_{\min}$).

Cette date est appelée date de début au plus tard de la tâche t et on la désigne par $\textit{tard}(t)$. La date de début au plus tard du sommet *fin* est fixée à $D_{\min}$.

Soient (t, x) un arc et C le chemin allant de t à *fin* et commençant par l'arc (t, x). L'exécution des tâches se trouvant sur le chemin C prend un temps égal au poids de C. La tâche t ne peut donc pas débuter après la date $D_{\min}$ moins le poids de C. Et puisque plusieurs tâches peuvent être postérieures à t, il vient que :

$$\textit{tard}(t) = D_{\min} - \Delta(t)$$

où $\Delta(t)$ est égal au poids d'un plus long chemin allant du sommet t au sommet *fin*

Bellman et calcul des dates de début au plus tard

L'algorithme de Bellman qui détermine **une anti-arborescence de chemins de poids maximum aboutissant au sommet *fin*** est le suivant :

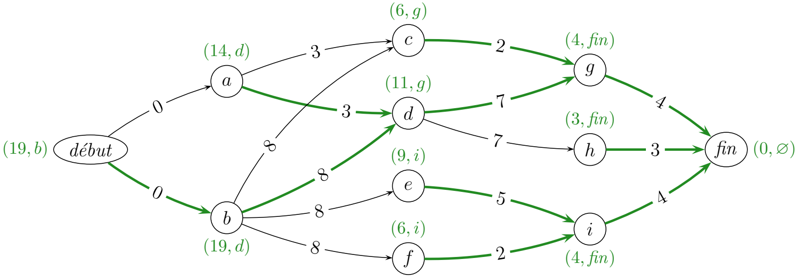
- **Initialisation des potentiels :** $\Delta[\mathit{fin}] = 0$ et $\Delta[t] = -1$ pour tout sommet $t \neq \mathit{fin}$.

- **Itérations :** pour chaque sommet t du graphe pris dans un **ordre topologique inverse**, calculer $\boldsymbol{\Delta[t] = \max_{x \in V^+(t)}\{\Delta(x) + p(t,x)\}}$. Cela revient à **relâcher tous les arcs d'origine t**, le relâchement d'un arc (t, x) s'écrivant ici :

  si $\Delta[x] + p(t,x) > \Delta[t]$ **alors**
  | $\Delta[t] \leftarrow \Delta[x] + p(t,x)\,;\ \mathcal{F}[t] \leftarrow x$
  fin si

- **Calcul des marges totales :** pour tout sommet t du graphe, calculer $\mathit{tard}(t) = D_{\min} - \Delta[t]$.

On utilise l'ordre topologique inverse ($fin, i, h, g, f, e, d, c, b, a, début$) :



t	fin	i	h	g	f	e	d	c	b	a	$début$
$\Delta(t)$	0	4	3	4	6	9	11	6	19	14	19
$fils$: $\mathcal{F}(t)$	$\varnothing$	fin	fin	fin	i	i	g	g	d	d	b
$\boldsymbol{tard(t)}$	**19**	**15**	**16**	**15**	**13**	**10**	**8**	**13**	**0**	**5**	**0**

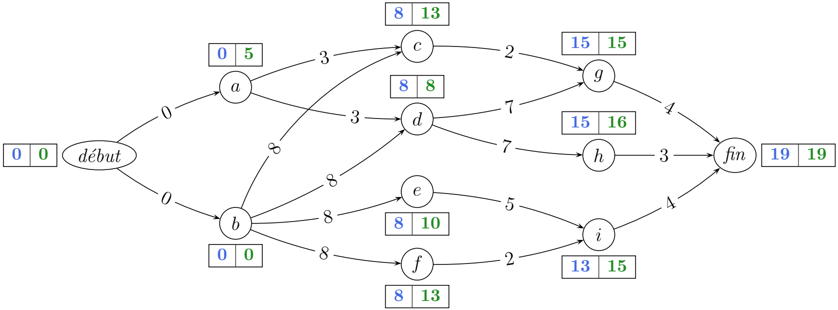
Le relâchement de tous les arcs d'origine $t \neq \mathit{fin}$ consiste à calculer :

$$\Delta(t) = \max_{x \in V^+(t)} \{ \ \Delta(x) + p(t, x) \ \}$$

Or $\mathit{tard}(t) = D_{\min} - \Delta(t)$, donc :

$$\begin{aligned}
D_{\min} - \mathit{tard}(t) &= \max_{x \in V^+(t)} \{ \ D_{\min} - \mathit{tard}(x) + \mathit{durée}(t) \ \} \\
\cancel{D_{\min}} - \mathit{tard}(t) &= \cancel{D_{\min}} + \max_{x \in V^+(t)} \{ \ -\mathit{tard}(x) + \mathit{durée}(t) \ \} \\
\mathit{tard}(t) &= - \max_{x \in V^+(t)} \{ \ -\mathit{tard}(x) + \mathit{durée}(t) \ \} \\
\boldsymbol{\mathit{tard}(t)} &= \boldsymbol{\min_{x \in V^+(t)} \{ \ \mathit{tard}(x) - \mathit{durée}(t) \ \}}
\end{aligned}$$

On peut écrire une version de l'algorithme de Bellman qui calcule directement les dates de début au plus tard. On pose initialement $\Delta(\mathit{fin}) = D_{\min}$. Puis, pour chaque sommet t pris dans ordre topologique inverse pour le graphe, on relâche tous les arcs d'origine t pour calculer $\Delta(t) = \min_{x \in V^+(t)}\{\Delta(x) - p(t, x)\}$. Et au final, $\mathit{tard}(t) = \Delta(t)$ pour toute tâche t.



$$\textit{tôt}(\textit{début}) = \textit{tard}(\textit{début}) = 0 \qquad \textit{tard}(\textit{début}) = \textit{tard}(\textit{fin}) = D_{\min}$$

$$\textit{tôt}(t) = \max_{x \in V^-(t)} \{\textit{tôt}(x) + \textit{durée}(x)\} \qquad \textit{tard}(t) = \min_{x \in V^+(t)} \{\textit{tard}(x) - \textit{durée}(t)\}$$

L'exécution de toute tâche t doit débuter entre les dates $\textit{tôt}(t)$ et $\textit{tard}(t)$ incluses pour respecter la durée minimale du projet.

Certaines tâches disposent d'une marge de manœuvre qui est sans conséquence sur la date de fin du projet (comme c qui peut commencer à la date 10 au lieu de la date 8). D'autres n'ont aucune marge de manœuvre (comme g).

Des événements imprévus peuvent retarder le démarrage d'une tâche. Des problèmes peuvent aussi survenir pendant l'exécution d'une tâche, nécessitant un délai supplémentaire pour la terminer. Ces deux situations (ainsi qu'une combinaison des deux) sont comparables et reviennent à considérer que la durée d'exécution d'une tâche t passe de $\mathit{durée}(t)$ à $\mathit{durée}(t) + r$, $r > 0$. On dira que t accuse un retard de r.

Il est donc utile de calculer la **marge de manœuvre** d'une tâche, c.-à-d. le **retard maximum** que peut prendre une tâche **sans que cela ne remette en question la date de fin au plus tôt du projet**.

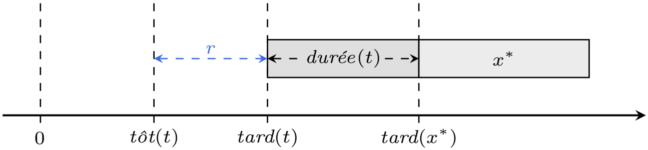
On distingue en fait 2 types de marge pour une même tâche t :

– La marge qui **ne modifie pas la date de début au plus tôt** des tâches suivant t, appelé **marge libre** de t.

– La marge qui **ne modifie pas la date de début au plus tard** des tâches suivant t, appelé **marge totale** de t.

Soit t une tâche. Rappelons que $\mathit{tard}(t) = \min_{x \in V^+(t)}\{\mathit{tard}(x) - \mathit{durée}(t)\}$ qui s'écrit aussi $\mathit{tard}(t) = \min_{x \in V^+(t)}\{\mathit{tard}(x)\} - \mathit{durée}(t)$.

Soit x^* la tâche de $V^+(t)$ ayant la plus petite date de début au plus tard. Alors $\mathit{tard}(t) = \mathit{tard}(x^*) - \mathit{durée}(t) \Leftrightarrow \mathit{tard}(t) + \mathit{durée}(t) = \mathit{tard}(x^*)$ et toute autre tâche x qui suit t vérifie $\mathit{tard}(x^*) \leq \mathit{tard}(x)$.



Si t accuse un retard au plus égal à $r = \mathit{tard}(t) - \mathit{tôt}(t)$ alors $\mathit{tard}(x^*)$ ne sera pas différée. Il en va de même pour toute autre tâche x suivant t puisque $\mathit{tard}(x^*) \leq \mathit{tard}(x)$.

Marge totale (2)

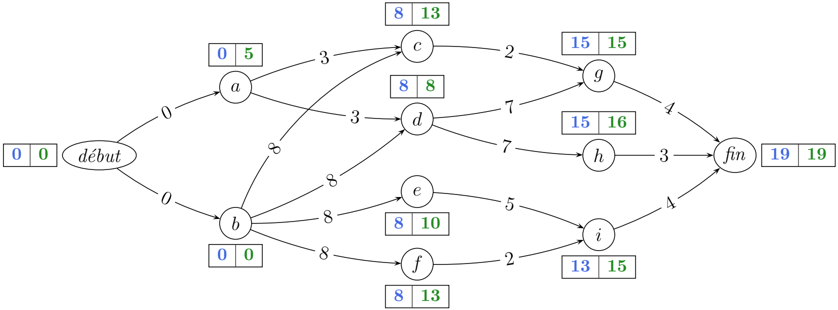
La marge totale d'une tâche t est :

$$M(t) = tard(t) - tôt(t)$$

Rappel : si une tâche t accuse un retard au plus égal à $M(t)$ alors la date fin au plus tôt du projet ne sera pas différée.

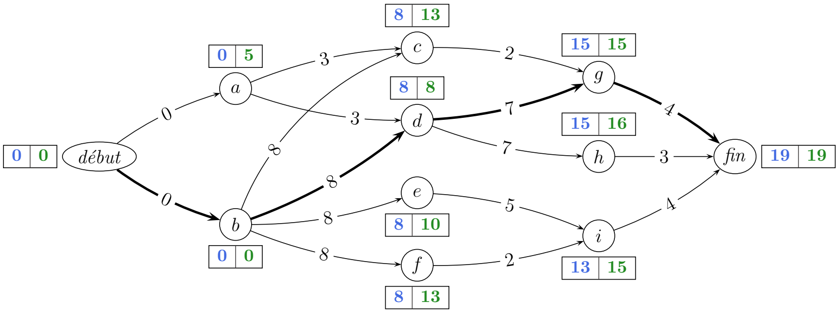
Si plusieurs tâches accusent un retard égal à leur marge totale respective alors la date fin au plus tôt du projet risque d'être différée.

Des retards égaux aux marges totales **ne peuvent pas être cumulés sans risque**.



t	*début*	a	b	c	d	e	f	g	h	i	*fin*
tôt(t)	0	0	0	8	8	8	8	15	15	13	19
tard(t)	0	5	0	3	8	10	13	15	16	15	19
$\boldsymbol{M(t)}$	**0**	**5**	**0**	**5**	**0**	**2**	**5**	**0**	**1**	**2**	**0**

Que se passe t-il si e accuse un retard de 2 ? Que se passe-t-il si e et i accusent chacune un retard de 2 ? Comment exploiter les marges totales ?



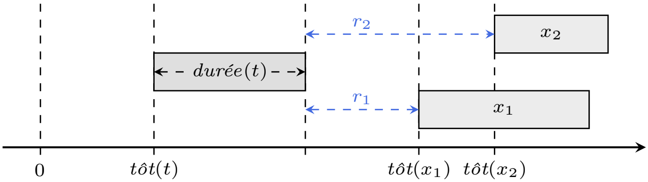
Si $\boldsymbol{tôt(t) = tard(t) \Leftrightarrow M(t) = 0}$ alors aucun retard n'est autorisé pour la tâche t qui est dite **critique**.

Un chemin critique est un chemin de poids maximum allant de *début* à *fin*. **Toute tâche critique appartient à un chemin critique**.

Les tâches critiques doivent donc être surveillées de près.

Marge libre (1)

Considérons une tâche t suivie par exactement 2 tâches, x_1 et x_2 :



Le retard maximum que peut prendre t sans que :

- $tôt(x_1)$ ne soit différée est $r_1 = tôt(x_1) - tôt(t) - durée(t)$,
- $tôt(x_2)$ ne soit différée est $r_2 = tôt(x_2) - tôt(t) - durée(t)$.

Si la tâche t accuse un retard au plus égal à $r_1 = tôt(x_1) - tôt(t) - durée(t)$ alors ni $tôt(x_1)$ ni $tôt(x_2)$ ne seront différées.

La marge libre d'une tâche t est :

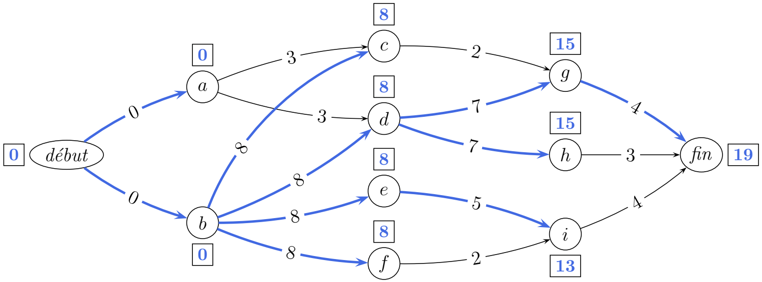
$$\begin{aligned} m(t) &= \min_{x \in V^+(t)} \left\{ tôt(x) - tôt(t) - durée(t) \right\} \\ &= \min_{x \in V^+(t)} \left\{ tôt(x) \right\} - tôt(t) - durée(t) \end{aligned}$$

Rappel : si une tâche t accuse un retard au plus égal à $m(t)$ alors la durée minimale du projet ne sera pas augmentée.

Si chaque tâche accuse un retard égal à sa marge libre alors la durée minimale du projet ne sera pas augmentée.

Des retards égaux aux marges libres **peuvent être cumulés sans aucun risque**.

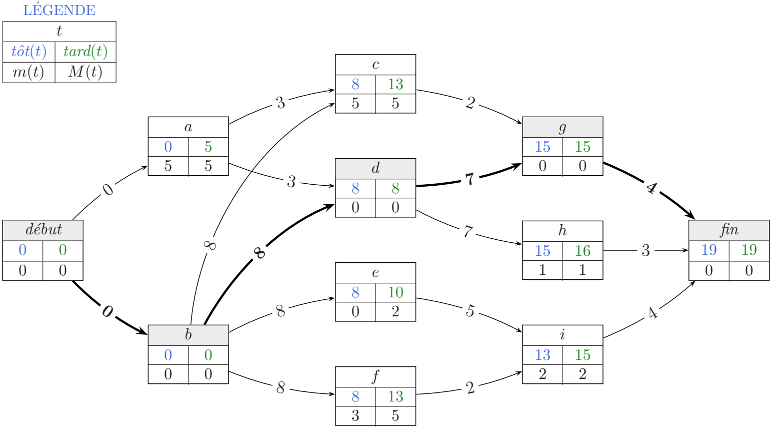
P. ex. : $m(a) = \min\{tôt(c)\text{-}tôt(a)\text{-}durée(a),\ tôt(d)\text{-}tôt(a)\text{-}durée(a)\} = \min\{8\text{-}0\text{-}3, 8\text{-}0\text{-}3\} = 5$. Ou : $m(a) = \min\{tôt(c), tôt(d)\}\text{-}tôt(a)\text{-}durée(a) = \min\{8, 8\}\text{-}0\text{-}3 = 5$.



t	*début*	a	b	c	d	e	f	g	h	i	*fin*
$tôt(t)$	0	0	0	8	8	8	8	15	15	13	19
$\boldsymbol{m(t)}$	**0**	**5**	**0**	**5**	**0**	**0**	**3**	**0**	**1**	**2**	**0**

Que se passe-t-il si f accuse un retard de 3 et i un retard de 2 ? Comment exploiter les marges libres ?

Résultat final résumé en un seul diagramme



Ce graphe est une représentation visuelle de la planification du projet. Il contient toutes les informations utiles au suivi et au contrôle du projet : tâches, durées, contraintes, dates, marges et taches critiques.