MPM : Méthode des Potentiels Métra

1. Exemple de données

Un projet est décomposé en les tâches suivantes :

tâches	a	b	b	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

Table 1 – Contraintes d'antériorité des tâches composant un projet

Il s'agit d'établir un calendrier d'exécution des tâches ou ordonnancement des tâches qui respecte les contraintes d'antériorité de chaque tâche. Une fois établi, ce calendrier permettra de suivre et de contrôler l'état d'avancement du projet.

Le problème consiste plus précisément à déterminer un calendrier d'exécution des tâches qui **minimise la durée globale d'exécution du projet**.

2. Graphe potentiel tâche

À partir des données (cf. Table 1), on construit le graphe potentiel tâche $G = (S, A, w)$ de la façon suivante :

- Chaque tâche est représentée par un sommet.
- Chaque contrainte d'antériorité de la forme « la tâche t_1 doit précéder la tâche t_2 » est représentée par un arc (t_1, t_2) de poids $w(t_1, t_2) = \textit{durée}(t_1)$. Tous les arcs sortant d'un même sommet auront donc le même poids.
- On ajoute un sommet noté *début* et des arcs de poids 0 allant de *début* vers les sommets représentant les tâches sans contrainte d'antériorité (comme a et b). Ce sommet représente une tâche fictive de durée 0 qui symbolise l'instant de début des travaux.
- On ajoute également un sommet noté *fin*. Pour chaque tâche x qui n'est suivie par aucune tâche (comme g, h et i), on ajoute un arc $(x, \textit{fin})$ de poids égal à $\textit{durée}(x)$. Ce sommet représente une tâche fictive qui symbolise l'instant de fin des travaux.

Le graphe ainsi obtenu est un DAG (cf. Figure 1) qui admet une unique source (le sommet *début*) et un unique puits (le sommet *fin*).

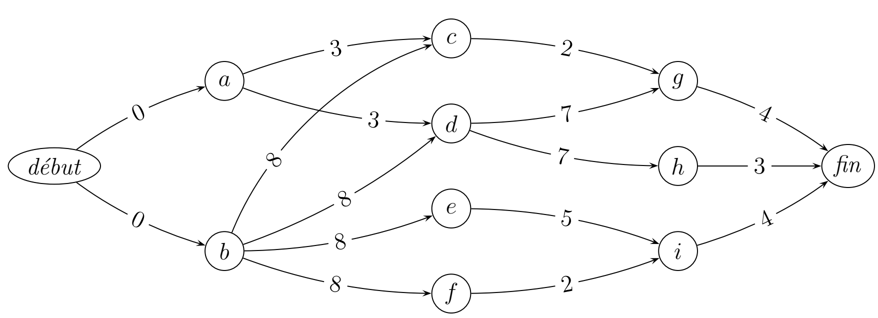

Figure 1 – Graphe potentiel tâche représentant le projet décrit dans l'exemple

3. Ordonnancement et ordre topologique

Tout ordre topologique pour un graphe potentiel tâche est un ordre d'exécution des tâches. Dans le cas de l'exemple, un ordre topologique possible est $(début, a, b, c, d, e, f, g, h, i, fin)$ et le calendrier d'exécution qui s'en déduit est le suivant : la tâche a démarre à la date 0, la tâche b à la date 3, la tâche c à la date 11, etc. En exécutant les tâches dans cet ordre, les contraintes d'antériorité seront respectées et la durée globale de réalisation du projet sera de 38.

Il est possible de réduire cette durée puisque certaines tâches peuvent être exécutées en parallèle. Par exemple, l'exécution de la tâche a et celle de la tâche b (ces deux tâches n'ont pas de contraintes d'antériorité) peuvent démarrer à la date 0. La durée globale passe alors de 38 à 35.

4. Objectif

L'objectif de la Méthode des Potentiels Métra (MPM) est de déterminer un calendrier d'exécution des tâches qui minimise la durée globale de réalisation du projet. Elle reprend les notions de dates et de marges comme définies dans la méthode PERT.

5. Dates de début au plus tôt et algorithme de Bellman

Soit t une tâche et soit x une tâche antérieure à t (autrement dit, (x, t) est un arc du graphe). Si l'exécution de x démarre à une date d alors l'exécution de t ne pourra pas démarrer avant la date $d + durée(x)$. D'où la notion de date de début au plus tôt d'une tâche t que l'on calcule en tenant compte de toutes les tâches qui précèdent la tâche t.

On appelle date de début au plus tôt d'une tâche t, le nombre $tôt(t)$ défini par :

$$tôt(début) = 0 \quad \text{et} \quad tôt(t) = \max_{x \in V^-(t)} \{ \; tôt(x) + durée(x) \; \} \tag{1}$$

où $V^-(t)$ représente l'ensemble des tâches antérieures à une tâche t.

Si l'exécution de chaque tâche t dure exactement $durée(t)$ alors la durée globale du projet, notée $D_{\min}$, sera égale à $tôt(fin)$. La réalisation du projet démarre donc à la date $tôt(début) = 0$ et se terminera au plus tôt à la date $tôt(fin) = D_{\min}$.

Soit T une arborescence de chemins de poids maximum de racine le sommet *début*. Soient la table des potentiels Π et la table des pères $\mathcal{P}$ représentant T. Alors, d'après l'équation (1) :

$$\forall t \in S, \; tôt(t) = \Pi(t)$$

En effet, puisque le poids d'un arc (x, t) est égal au poids de la tâche x, l'équation (1) s'écrit aussi : $tôt(t) = \max_{x \in V^-(t)}\{tôt(x) + w(x, t)\}$. La date de début au plus tôt d'une tâche t est donc obtenue en relâchant tous les arcs de but t (cf. Figure 2 et Table 2).

6. Dates de début au plus tard et algorithme de Bellman

Considérons une tâche t qui n'est suivie par aucune autre tâche. L'exécution de t devra alors commencer au plus tard à la date $D_{min} - dure(t)$ sinon la date de fin du projet sera repoussée.

Considérons une tâche t et soit x une tâche postérieure à la tâche t (autrement dit, (t, x) est un arc du graphe). Supposons que l'exécution de x doivent démarrer au plus tard à une date d pour ne pas retarder la suite des travaux. Alors l'exécution de la tâche t ne pourra pas démarrer après la date $d - durée(t)$. D'où la notion de date de début au plus tard d'une tâche t que l'on calcule en tenant compte de toutes les tâches qui suivent la tâche t.

On appelle date de début au plus tard d'une tâche t, le nombre $tard(t)$ défini par :

$$\boldsymbol{tard(fin) = D_{\min} \quad \text{et} \quad tard(t) = \min_{x \in V^+(t)} \{\ tard(x) - durée(t)\ \}} \qquad (2)$$

où $V^+(t)$ représente l'ensemble des tâches qui suivent la tâche t.

Poser $tard(fin) = D_{\min}$ permet de fixer la durée globale du projet à $D_{\min}$. Puis chaque date $tard(t)$ est calculée en fonction des dates de début au plus tard des successeurs de t dans le graphe. De cette façon, $tard(t)$ devient la date à laquelle l'exécution de t peut débuter sans que cela ne repousse la date de début au plus tard de toutes les tâches qui suivent t et donc sans repousser la date de fin du projet.

Soit T une anti-arborescence de chemins de poids maximum d'anti-racine le sommet *fin*. Soient la table des potentiels Δ et la table des fils $\mathcal{F}$ représentant T. Alors, d'après l'équation (2) :

$$\boldsymbol{\forall t \in S,\ tard(t) = D_{\min} - \Delta(t)}$$

En effet, si on remplace $tard(.)$ par $D_{\min} - \Delta(.)$ dans l'équation (2), on obtient :

$$tard(fin) = D_{\min} \Leftrightarrow D_{\min} - \Delta(fin) = D_{\min} \Leftrightarrow \boldsymbol{\Delta(fin) = 0}$$

Et pour chaque tâche $t \neq fin$:

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

Ce qui montre que $\Delta(t)$ est calculé en relâchant tous les arcs d'origine t.

Supposons que les durées des tâches soient exprimées en jours. Les potentiels calculés par l'algorithme de Bellman s'interprètent alors comme suit : la date de fin du projet est fixé à $\Delta(fin) = 0$ et une tâche $t \neq fin$ doit commencer au plus tard $\Delta(t)$ jours avant la date de fin pour que celle-ci soit respectée (voir Figure 3 et Table 3). Notons que l'on obtient le même résultat en remplaçant les durées par leur opposé et en utilisant l'algorithme de Bellman pour rechercher les chemins de poids minimum aboutissant au sommet *fin*.

7. Marge libre

Si l'exécution d'une tâche t démarre à la date $tôt(t)$ alors elle se terminera à la date $tôt(t) + durée(t)$ qui est donc la date de fin au plus tôt de la tâche t. Considérons une tâche x qui suit la tâche t : si l'exécution de t accuse un retard d'au plus $tôt(x) - (tôt(t) + durée(t))$ alors l'exécution de x pourra tout de même commencer à la date $tôt(x)$. D'où la notion de marge libre d'une tâche t que l'on calcule en tenant compte toutes les tâches qui suivent t.

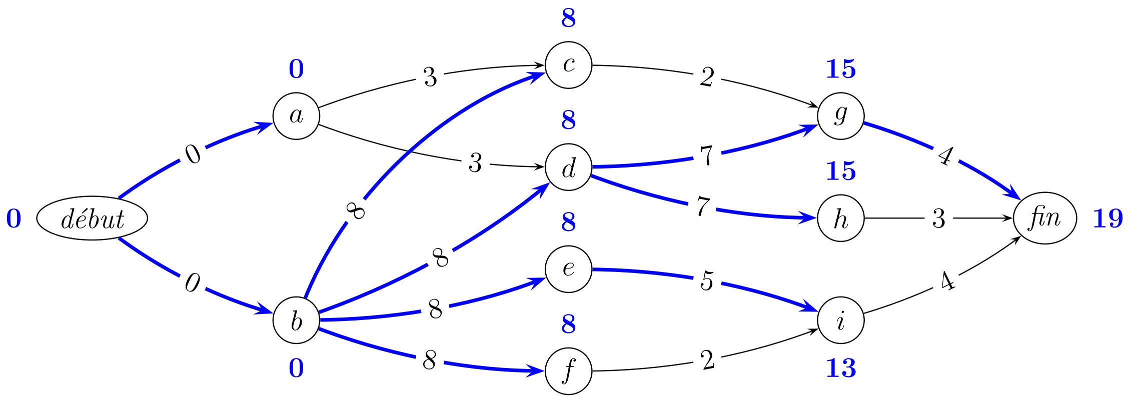

FIGURE 2 – Arborescence obtenue par algorithme de Bellman

t	*début*	a	b	c	d	e	f	g	h	i	*fin*
$\textit{tôt}(\boldsymbol{t}) = \boldsymbol{\Pi(t)}$	**0**	**0**	**0**	**8**	**8**	**8**	**8**	**15**	**15**	**13**	**19**
père : $\mathcal{P}(t)$	none	*début*	*début*	b	b	b	b	d	d	i	g

TABLE 2 – Dates de début au plus tôt

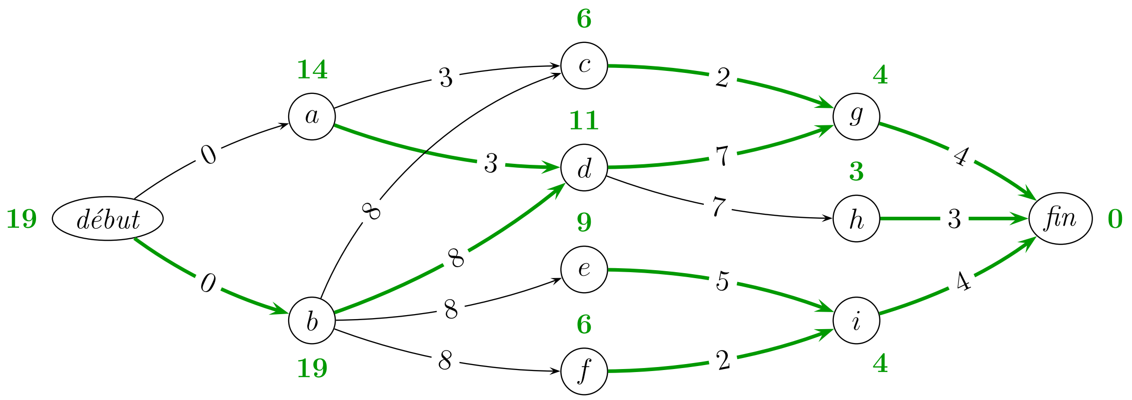

FIGURE 3 – Anti-arborescence obtenue par l'algorithme de Bellman

t	*début*	a	b	c	d	e	f	g	h	i	*fin*
$\Delta(t)$	19	14	19	6	11	9	6	4	3	4	0
fils : $\mathcal{F}(t)$	b	d	d	g	g	i	i	*fin*	*fin*	*fin*	none
$\boldsymbol{tard(t) = 19 - \Delta(t)}$	**0**	**5**	**0**	**13**	**8**	**10**	**13**	**15**	**16**	**15**	**19**

TABLE 3 – Dates de début au plus tard

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	13	8	10	13	15	16	15	19
$\boldsymbol{m(t)}$	**0**	**5**	**0**	**5**	**0**	**0**	**3**	**0**	**1**	**2**	**0**
$\boldsymbol{M(t)}$	**0**	**5**	**0**	**5**	**0**	**2**	**5**	**0**	**1**	**2**	**0**

TABLE 4 – Dates et marges

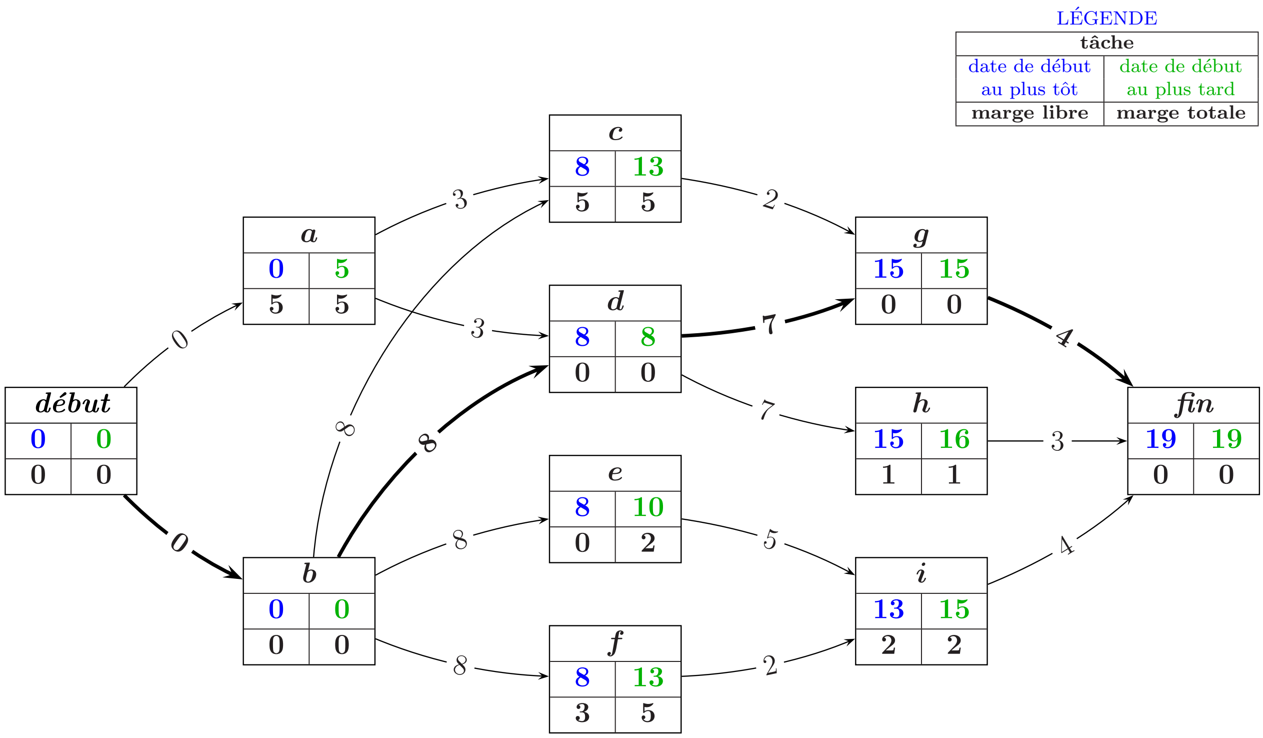

FIGURE 4 – MPM en un seul diagramme

La marge libre d'une tâche t est le retard maximal que peut prendre la réalisation de t sans que cela ne repousse la date de début au plus tôt de toute tâche qui suit t. On la note $m(t)$ et :

$$\boldsymbol{m(t) = \min_{x \in V^+(t)} \{\ t\hat{o}t(x) - (t\hat{o}t(t) + dur\acute{e}e(t))\ \}} \tag{3}$$

où de manière équivalente :

$$\boldsymbol{m(t) = \min_{x \in V^+(t)} \{\ t\hat{o}t(x)\ \} - t\hat{o}t(t) - dur\acute{e}e(t)}$$

Supposons que chaque tâche soit confiée à une personne différente. Celle qui est responsable de la tâche t commencera l'exécution de t à la date $tôt(t)$, si toutefois les tâches qui précèdent t se sont terminées sans retard. Si tout ce passe bien, l'exécution de t se terminera alors au plus tôt à la date $tôt(t) + durée(t)$. En cas de problème, la personne dispose d'une marge d'au plus $m(t)$ pour finaliser l'exécution de t sans que cela n'impacte la suite du projet. Il n'est pas nécessaire d'avertir les personnes en charge des tâches qui suivent t puisque l'exécution de toute tâche x suivant t pourra commencer comme prévu à la date $tôt(x)$.

Calculons par exemple la marge libre de a (cf. Table 4 et Figure 4). On a $V^+(a) = \{c, d\}$ et $tôt(a) + durée(a) = 0 + 3 = 3$ d'où $m(a) = \min\{tôt(c), tôt(d)\} - 3 = \min\{8, 8\} - 3 = 5$. Un retard de 5 au maximum est donc autorisé lors de la réalisation de la tâche a sans que cela ne modifie la date de début au plus tôt des taches c et d.

Si chaque tâche accuse un retard inférieur ou égal à sa marge libre, la date de fin du projet ne sera pas modifiée. Des retards au plus égaux aux marges libres peuvent donc être cumulés sans risque.

Considérons maintenant le cas de la tâche e qui a une marge libre de 0 : si l'exécution de e accuse un retard de 1, cela va repousser de 1 la date de début au plus tôt de la tâche i sans pour autant repousser la date de début au plus tard de i et donc sans repousser la date de fin du projet. Il existe donc un délai supplémentaire pour cette tâche e qui, s'il est utilisé, n'impactera pas la durée globale du projet.

8. Marge totale

On définit la marge totale $M(t)$ d'une tâche t par :

$$\boldsymbol{M(t) = tard(t) - t\hat{o}t(t) \quad \text{et} \quad M(t) \geq m(t)} \tag{4}$$

La marge totale d'une tâche t est le retard maximal que peut prendre la réalisation de t sans que cela ne repousse la date de début au plus tard des tâches qui suivent t et donc sans que cela ne repousse la date de fin du projet.

Par exemple (cf. Table 4 et Figure 4), $M(e) = tard(e) - tôt(e) = 2 - 0 = 2$: un retard de 2 au maximum est autorisé lors de l'exécution de la tâche e sans que cela ne modifie la date de début au plus tard de la tâche i ni la date de fin du projet. Mais si en plus de la tâche e, la tâche i accuse aussi un retard de $M(i) = 2$ alors la fin date de fin du projet passera de 19 à 21. En ce qui concerne la tâche b de marge totale $M(b) = 0$, tout retard dans sa réalisation repoussera d'autant la date de fin du projet.

Si plusieurs tâches (appartenant à un même chemin) accusent un retard égal à leur marge totale, la date de fin du projet peut être repoussée. Des retards égaux aux marges totales ne peuvent donc pas être cumulés sans risque.

Supposons que la personne en charge de la tâche t commence l'exécution de t à la date $tôt(t)$ comme prévu. Si tout ce passe bien, l'exécution se terminera alors au plus tôt à la date $tôt(t) + durée(t)$. En cas de problème, la personne dispose d'une marge de manoeuvre d'au plus $M(t)$ pour finaliser l'exécution de t. Cependant, si elle utilise l'intégralité de cette marge, elle devra en informer les personnes en charge des tâches qui suivent t car leur date de début au plus tôt peuvent être impactées (mais pas leur date de début au plus tard).

Considérons le cas de la tâche f (cf. Table 4 et Figure 4). Si l'exécution de f à un retard d'au plus $m(f) = 3$, la date de début au plus tôt de i ne changera pas et restera à 13. Un retard de 4 va repousser à 14 la date de début au plus tôt de i. Un retard de $M(f) = 5$ va repousser à 15 la date de début au plus tôt de i. Dans toutes les situations décrites ci-dessus, la date de fin du projet reste inchangée. La durée globale du projet ne sera allongée que si le retard de f est strictement supérieur à $M(f)$.

9. Tâches critiques

Une tâche t est dite critique si sa marge totale est nulle, c.-à-d. si $M(t) = 0$. Tout retard dans la réalisation d'une tâche critique a pour conséquence d'augmenter la durée globale d'exécution du projet. L'exécution des tâches critiques doit donc faire l'objet d'une attention particulière.

Dans le cas de l'exemple (voir Table 4 et Figure 4), les tâches critiques sont les tâches b, d et g.

10. Chemin critique

Un chemin critique est un chemin de poids maximum allant du sommet *début* au sommet *fin* dans le graphe potentiel tâche. Tout chemin critique est exclusivement composé de tâches critiques.

Dans le cas de l'exemple, il n'existe qu'un seul chemin critique : $(début, b, d, g, fin)$.