7B La représentation des tâches par les sommets: le réseau potentiels-tâches

Dans le chapitre 7, les projets ont été représentés graphiquement par des réseaux dont les arcs correspondent aux tâches et dans lesquels, pour indiquer que la tâche S est un prédécesseur immédiat de T, on fait coïncider le sommet initial de T avec le sommet terminal de S. Il existe une autre approche où ce sont les sommets qui correspondent aux tâches. Les anglo-saxons parlent alors de *Activity On Node (AON) Graph*. Bernard Roy nomme **graphe potentiels-tâches** une telle représentation graphique. Par opposition, il qualifie de **graphe potentiels-étapes** la représentation plus traditionnelle décrite dans les sections 7.1 et 7.2.

La construction d'un réseau potentiels-tâches

Considérons, à titre d'exemple, le projet abstrait dont les tâches sont énumérées au tableau 1 (il s'agit du même exemple déjà décrit au tableau 7.2 et dont le réseau potentiels-étapes a été construit dans la section 7.2).

TABLEAU 1. Un projet abstrait: description

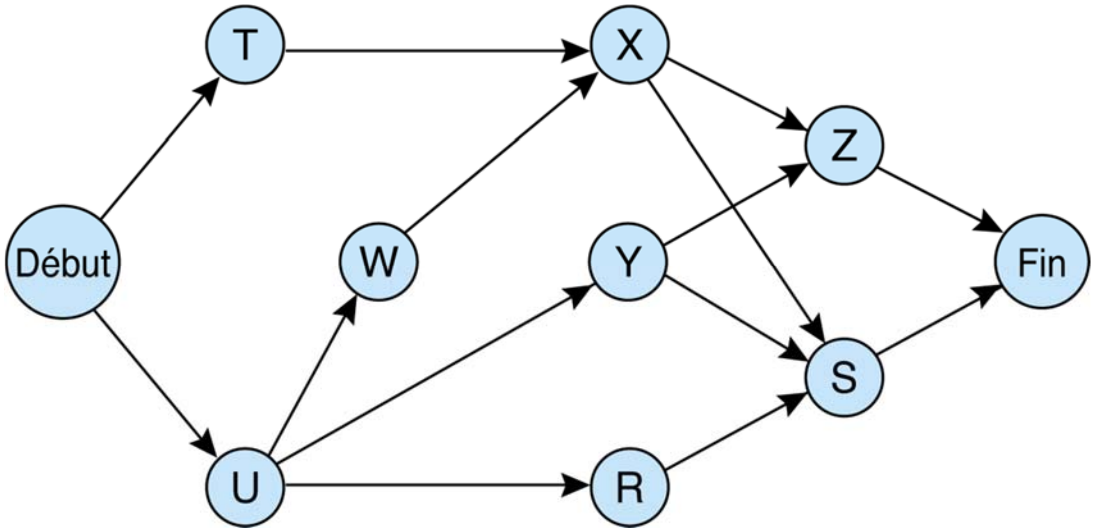
Tâche	Prédécesseur(s) immédiat(s)	Durée
T	–	7
U	–	4
W	U	4
X	T, W	5
Y	U	6
Z	X, Y	8
R	U	4
S	X, Y, R	6

La figure 1 donne le réseau potentiels-tâches associé à ce projet. Trois règles ont présidé à l'élaboration de cette représentation graphique.

- une tâche est représentée par un sommet ;
- quand une tâche *t* est prédécesseur immédiat de *t'*, on trace un arc du sommet associé à *t* vers celui correspondant à *t'*;
- on ajoute des sommets Début et Fin: le premier précède tout sommet n'admettant aucun prédécesseur immédiat, tandis que Fin est lié comme sommet terminal à tout sommet qui ne possède aucun successeur immédiat.

En principe, on peut disposer les sommets à son gré. Cependant, on essaie généralement de placer autant que possible un prédécesseur à gauche de ses successeurs, de minimiser le nombre de croisements des arcs. Ces conventions, qui souffrent un certain nombre d'exceptions, améliorent la lisibilité et l'élégance du réseau.

FIGURE 1. Un exemple abstrait: le réseau potentiels-tâches



Le calcul des moments et des marges

Le réseau d'un projet permet au gestionnaire de visualiser les relations entre les différentes tâches à réaliser pour mener ce projet à bonne fin. Il sert également de point de départ à la mise en œuvre de divers algorithmes reliés soit à l'ordonnancement optimal des tâches, soit au déroulement optimal du projet. Parmi ces algorithmes, expliquons d'abord celui qui permet d'établir la durée minimale d'un projet, sous l'hypothèse que la durée de chaque tâche est fixe et connue à l'avance.

À nouveau, le projet abstrait du tableau 1 servira d'illustration. En un premier temps, on calcule, pour toute tâche *t*, les moments au plus tôt ES(t) et EF(t).

- ES(*t*), le **début au plus tôt** de *t*, est par définition le nombre minimal de périodes entre le démarrage du projet et le début de la tâche *t*;
- EF(*t*), la **fin au plus tôt** de *t*, représente le délai minimal entre le démarrage du projet et le parachèvement de *t*.

Les notations ES(*t*) et EF(*t*) reflètent la terminologie anglo-saxonne, *Earliest Start* et *Earliest Finish*. Les fins au plus tôt s'obtiennent de la façon suivante :

$$\text{EF}(t) = \text{ES}(t) + (\text{durée de } t).$$

Il reste donc à calculer les débuts au plus tôt. Il résulte immédiatement de la définition de ES(*t*) que

$$\text{ES(Début)} = \text{ES(T)} = \text{ES(U)} = 0.$$

De plus, ES(W) = EF(U) + 4 = 0 + 4 = 4, car W peut commencer dès que la tâche U est complétée et cette dernière exige 4 périodes. On obtient de même que ES(Y) = ES(R) = 4.

Tout semble baigner dans l'huile. Déterminons donc ES(S): puisque X, Y et R sont les prédécesseurs immédiats de S, ces trois tâches devront être complétées avant que S ne puisse démarrer. Ainsi,

$$\mathrm{ES(S)} \geq \mathrm{EF(X)} = \mathrm{ES(X)} + 50$$

$$\mathrm{ES(S)} \geq \mathrm{EF(Y)} = 4 + 6 = 10$$

$$\mathrm{ES(S)} \geq \mathrm{EF(R)} = 4 + 4 = 8.$$

On constate que le calcul de ES(S) achoppe sur l'ignorance dans laquelle nous sommes temporairement de la valeur de ES(X). Mais X admet T et W comme prédécesseurs immédiats et, par conséquent,

$$\mathrm{ES(X)} \geq \mathrm{EF(T)} = 0 + 7 = 7$$

$$\mathrm{ES(X)} \geq \mathrm{EF(W)} = 4 + 4 = 8.$$

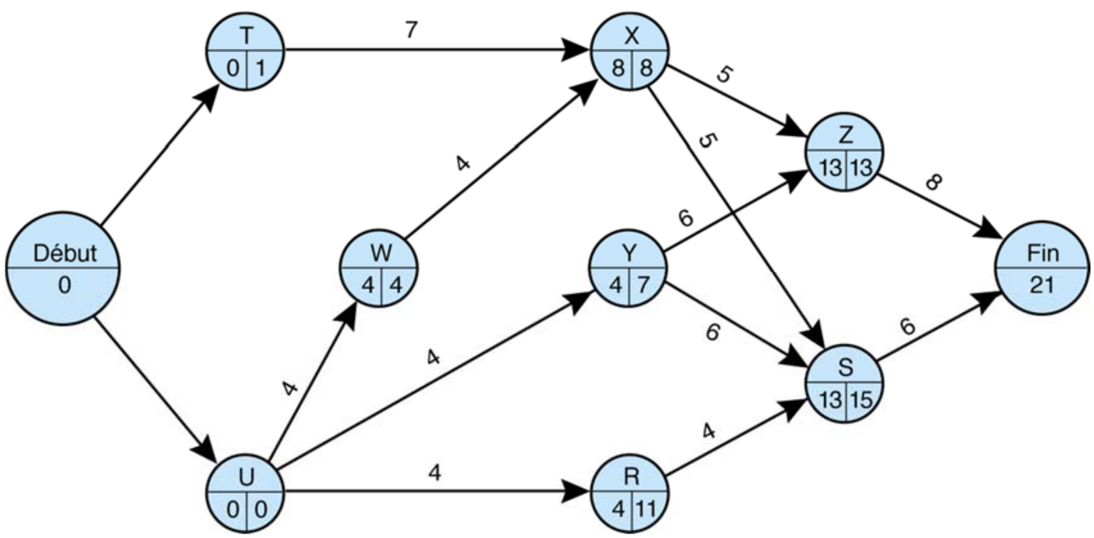
On en conclut que ES(X) = 8, puis que ES(S) = 13. On obtient enfin que

$$\mathrm{ES(Z)} = \max \{ 8 + 5; 4 + 6 \} = 13$$

$$\mathrm{ES(Fin)} = \max \{ 13 + 8; 13 + 6 \} = 21.$$

Ainsi, le délai minimal entre le démarrage du projet et la pseudo-tâche Fin, qui en représente la fin, est de 21 périodes. C'est la **durée minimale du projet**.

FIGURE 2. Un exemple abstrait: débuts au plus tôt et au plus tard



L'exemple précédent a montré que l'ordre dans lequel se calculent les valeurs des ES(t) n'est pas complètement arbitraire; en fait, le début au plus tôt d'un sommet ne peut être déterminé que si les débuts au plus tôt de tous ses prédécesseurs immédiats sont connus. Nous décrivons maintenant un **algorithme qui donne les moments au plus tôt** ES(t) et EF(t) des différentes tâches t d'un réseau potentiels-tâches. Afin d'en simplifier la présentation, nous convenons que les pseudo-tâches Début et Fin sont de durée nulle.

- Poser: ES(Début) = EF(Début) = 0.
- Choisir au hasard une tâche parmi celles dont les valeurs ES et EF de tous les prédécesseurs immédiats sont connues. Soient t, la tâche retenue; t_1, ..., t_k, ses prédécesseurs immédiats. Poser

$$ES(t) = \max \{EF(t_1); EF(t_2); ...; EF(t_k)\} \qquad (1)$$

$$EF(t) = ES(t) + (\text{durée de } t). \qquad (2)$$

Le calcul des moments au plus tôt s'effectue le plus souvent grâce à un logiciel. On peut également les calculer directement sur le réseau, pourvu que la durée de chaque tâche y ait été reportée. La figure 2 donne les valeurs ES(t) des différentes tâches du projet abstrait. Les valeurs dans les cases de droite représentent des débuts au plus tard LS(t), dont nous verrons ci-dessous l'interprétation et l'intérêt.

La tâche S pourrait s'étaler sur plus de temps que prévu au tableau 1 sans que la fin du projet n'en soit nécessairement retardée: en effet, le délai entre les sommets S et F est de 21 – 13 = 8 périodes, alors que la tâche S en exige 6 seulement. Cet excédent de 2 périodes, qui est qualifié de **marge**, représente le nombre maximal de périodes supplémentaires qu'on pourrait accorder à la tâche S sans que soit retardé le parachèvement du projet. On qualifie de **critiques** les **tâches** dont la marge est nulle: tout prolongement de leur durée retarde donc le parachèvement du projet.

Nous indiquons maintenant comment trouver les tâches critiques et calculer la marge positive des tâches non critiques. Mais tout d'abord introduisons, pour toute tâche t, les **moments au plus tard** LS(t) et LF(t). La **fin au plus tard** LF(t), *Latest Finish* en anglais, est par définition l'instant le plus tardif où devra être complétée la tâche t si la durée du projet doit rester minimale. De même, LS(t) représente l'instant le plus tardif pour commencer t si l'on veut éviter de retarder la fin du projet. Évidemment,

$$LS(t) = LF(t) – (\text{durée de } t).$$

Il reste donc à déterminer la fin au plus tard des différentes tâches. On procède en suivant à rebours l'ordre utilisé dans le calcul des moments au plus tôt. Il résulte immédiatement des définitions que LF(Fin) = LS(Fin) = 21 et que

$$LF(Z) = 21 \quad \text{et} \quad LS(Z) = 21 – 8 = 13$$

$$LF(S) = 21 \quad \text{et} \quad LS(S) = 21 – 6 = 15.$$

La tâche X, qui est prédécesseur immédiat de Z et de S, doit être complétée suffisamment tôt pour que ni le début de Z, ni celui de S ne soient retardés: par conséquent,

$$LF(X) \leq LS(Z) = 13 \quad \text{et} \quad LF(X) \leq LS(S) = 15.$$

Il en résulte que LF(X) = 13 et LS(X) = 13 – 5 = 8 . De même,

$$LF(Y) = \min \{LS(Z); LS(S)\} = 13 \quad \text{et} \quad LS(Y) = 13 – 6 = 7.$$

Enfin,

$$LF(R) = LS(S) = 15 \quad \text{et} \quad LS(R) = 15 – 4 = 11$$

$$LF(W) = LS(X) = 18 \quad \text{et} \quad LS(W) = 18 – 4 = 14$$

$$LF(T) = LS(X) = 18 \quad \text{et} \quad LS(T) = 18 - 7 = 11.$$

La tâche U admet trois successeurs immédiats W, Y et R: par conséquent,

$$LF(U) = \min \{LS(W); LS(Y); LS(R)\} = 4 \quad \text{et} \quad LS(U) = 4 - 4 = 0.$$

Voici un algorithme qui systématise la procédure suivie ci-dessus et qui donne les moments au plus tard LF(t) et LS(t) des différentes tâches t d'un réseau potentiels-tâches.

- Poser: LF(Fin) = LS(Fin) = durée minimale du projet.
- Choisir au hasard une tâche parmi celles dont les valeurs LF et LS de tous les successeurs immédiats sont connues. Soient t, la tâche retenue; t_1, ..., t_k, ses successeurs immédiats. Poser

$$LF(t) = \min \{LS(t_1); LS(t_2); ...; LS(t_k)\} \quad (3)$$

$$LS(t) = LF(t) - (\text{durée de } t). \quad (4)$$

Enfin, la marge d'une tâche se calcule conformément à la formule (8) de la section 7.3 :

$$\text{marge de } t \;=\; LS(t) - ES(t) \;=\; LF(t) - EF(t). \quad (5)$$

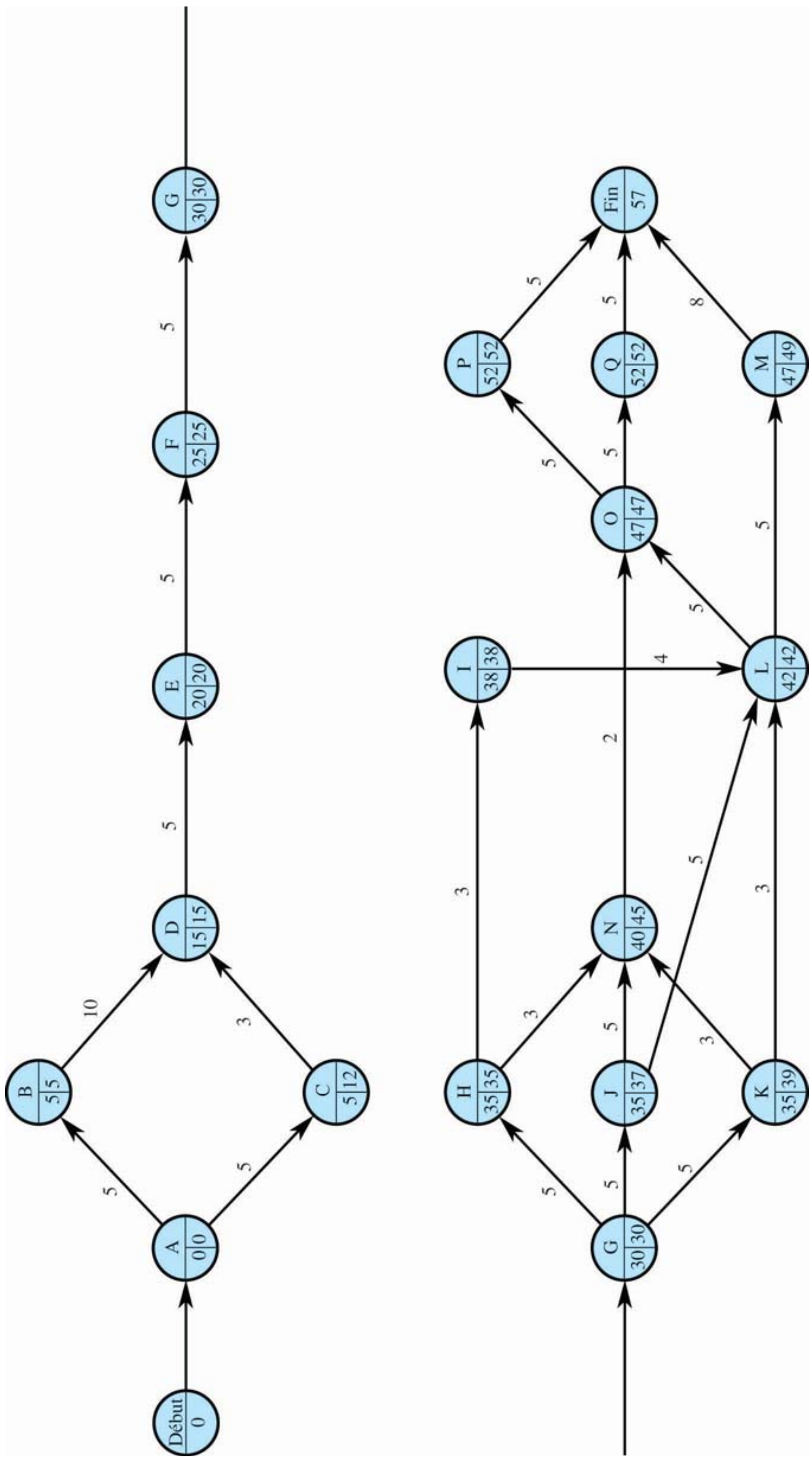
Le tableau 2 énumère les moments et les marges des différentes tâches du projet abstrait. La figure 2 décrit ce même projet par un réseau potentiels-tâches sur lequel ont été reportés les débuts au plus tôt et au plus tard.

TABLEAU 2. Un projet abstrait: moments et marges

Tâche	Début au + tôt	Fin au + tôt	Début au + tard	Fin au + tard	Marge
T	0	7	1	8	1
U	0	4	0	4	0
W	4	8	4	8	0
X	8	13	8	13	0
Y	4	10	7	13	3
Z	13	21	13	21	0
R	4	8	11	15	7
S	13	19	15	21	2

Enfin, le projet RESO présenté au tableau 7.1 est illustré à la figure 3 : on constate que l'implantation du réseau de micro-informatique exige 57 jours ouvrables, un résultat déjà trouvé aux figures 7.2 et 7.3 à partir de l'approche potentiels-étapes.

FIGURE 3. Projet RESO: le réseau potentiels-tâches



La compression de la durée d'un projet

Lorsqu'il s'agit de fixer la durée d'une tâche, la coutume veut que de toutes les durées possibles l'on retienne celle qui s'avérerait la moins onéreuse. En y consacrant les ressources nécessaires, il est possible, si le projet revêt un caractère urgent, de diminuer la durée de certaines des tâches qui le constituent. On parlera alors d'accélérer ces tâches pour compresser la durée du projet.

Indiquons comment analyser de telles situations. Reprenons l'exemple abstrait décrit au tableau 1. Les tâches dont l'accélération est possible et les coûts afférents si une telle décision était prise sont donnés au tableau 3 : par exemple, la durée de la tâche T pourrait être réduite de 7 à 4 périodes mais, en contrepartie, on devrait alors débourser 10 000 $ au lieu des 7 000 $ qu'il en coûterait pour exécuter T conformément au programme normal. On convient généralement que la durée effective d'une tâche peut être fixée à toute valeur comprise entre les durées normale et accélérée et que les coûts évoluent **linéairement** en fonction de cette durée effective: par exemple, ceux-ci s'élèveraient à 9 000 $ si l'on décidait d'exécuter T en 5 périodes.

TABLEAU 3. Un projet abstrait: coûts d'accélération des tâches

Tâche	Programme normal		Programme accéléré		Coût d'accélération (en $ / période)
	Durée	Coût (en $)	Durée	Coût (en $)	
T	7	7 000	4	10 000	1 000
U	4	20 000	4	20 000	–
W	4	4 000	2	8 000	2 000
X	5	10 000	4	14 000	4 000
Y	6	12 000	4	17 000	2 500
Z	8	8 000	5	14 000	2 000
R	4	8 000	3	11 000	3 000
S	6	18 000	5	21 000	3 000

Nous avons déjà calculé que la **durée normale** du projet, c'est-à-dire la durée minimale sans compression d'aucune tâche, est de 21 périodes. Si l'on désirait ramener cette durée à 18 périodes, il faudrait accélérer certaines tâches. Mais lesquelles choisir et de combien de périodes accélérer chacune, tout en minimisant les coûts additionnels? La réponse à cette question s'obtient comme solution d'un modèle linéaire continu.

Mais auparavant, nous construisons un modèle linéaire plus simple, qui cherche seulement à calculer la durée minimale du projet, en l'absence de toute compression. Le modèle comprend les variables de décision suivantes :

Déb(t) = instant où débute la tâche t

DuréeP = durée du projet.

L'objectif consiste ici à minimiser la durée normale du projet :

$$\text{Min } z = \text{DuréeP}.$$

Les contraintes technologiques de ce modèle se regroupent en 2 catégories.

- Si t est prédécesseur immédiat de t', alors t' ne peut démarrer avant la fin de t :

$$\text{Déb}(t') \geq \text{Déb}(t) + \text{d}(t). \tag{6}$$

- Si t n'admet aucun successeur, alors

$$\text{DuréeP} \geq \text{Déb}(t) + \text{d}(t). \tag{7}$$

Dans le cas du projet abstrait, le modèle linéaire comprend 12 contraintes technologiques, dont voici la liste.

U AVANT W	:	– Déb(U) + Déb(W) ≥ 4
T AVANT X	:	– Déb(T) + Déb(X) ≥ 7
W AVANT X	:	– Déb(W) + Déb(X) ≥ 4
U AVANT Y	:	– Déb(U) + Déb(Y) ≥ 4
X AVANT Z	:	– Déb(X) + Déb(Z) ≥ 5
Y AVANT Z	:	– Déb(Y) + Déb(Z) ≥ 6
U AVANT R	:	– Déb(U) + Déb(R) ≥ 4
X AVANT S	:	– Déb(X) + Déb(S) ≥ 5
Y AVANT S	:	– Déb(Y) + Déb(S) ≥ 6
R AVANT S	:	– Déb(R) + Déb(S) ≥ 4
Z AVANT FIN	:	– Déb(Z) + DuréeP ≥ 8
S AVANT FIN	:	– Déb(S) + DuréeP ≥ 6

Il est facile d'adapter le modèle linéaire précédent pour déterminer quelles tâches accélérer et de combien de périodes chacune, de façon à parachever le projet dans les délais impartis tout en minimisant les coûts d'accélération. Tout d'abord, on ajoute au modèle des variables Acc(t) :

$$\text{Acc}(t) = \text{réduction, grâce à une accélération, de la durée de } t$$

où t est une tâche qui peut être accélérée. On adapte la fonction z au nouvel objectif de minimiser les coûts d'accélération. Ensuite, si t est une tâche qui peut être accélérée, on remplace, dans les inéquations (6) et (7), la durée normale d(t) de t par sa durée accélérée d(t) – Acc(t) :

si t est prédécesseur immédiat de t', alors $\text{Déb}(t') \geq \text{Déb}(t) + \mathbf{(}\text{d}(t) - \text{Acc}(t)\mathbf{)}$ (8)

si t n'admet aucun successeur, alors $\text{DuréeP} \geq \text{Déb}(t) + \mathbf{(}\text{d}(t) - \text{Acc}(t)\mathbf{)}$. (9)

Enfin, on doit tenir compte des bornes supérieures pour les variables DuréeP et Acc(t) qui découlent du contexte. Dans le cas des données du tableau 3, le modèle modifié prend la forme suivante. L'objectif est de minimiser le coût total z, où

$$z = \text{Acc(T)} + 2\ \text{Acc(W)} + 4\ \text{Acc(X)} + 2{,}5\ \text{Acc(Y)} + 2\ \text{Acc(Z)} + 3\ \text{Acc(R)} + 3\ \text{Acc(S)},$$

sous les contraintes technologiques suivantes.

U AVANT W	:	$-\text{Déb(U)} + \text{Déb(W)} \geq 4$
T AVANT X	:	$-\text{Déb(T)} + \text{Déb(X)} + \text{Acc(T)} \geq 7$
W AVANT X	:	$-\text{Déb(W)} + \text{Déb(X)} + \text{Acc(W)} \geq 4$
U AVANT Y	:	$-\text{Déb(U)} + \text{Déb(Y)} \geq 4$
X AVANT Z	:	$-\text{Déb(X)} + \text{Déb(Z)} + \text{Acc(X)} \geq 5$
Y AVANT Z	:	$-\text{Déb(Y)} + \text{Déb(Z)} + \text{Acc(Y)} \geq 6$
U AVANT R	:	$-\text{Déb(U)} + \text{Déb(R)} \geq 4$
X AVANT S	:	$-\text{Déb(X)} + \text{Déb(S)} + \text{Acc(X)} \geq 5$
Y AVANT S	:	$-\text{Déb(Y)} + \text{Déb(S)} + \text{Acc(Y)} \geq 6$
R AVANT S	:	$-\text{Déb(R)} + \text{Déb(S)} + \text{Acc(R)} \geq 4$
Z AVANT FIN	:	$-\text{Déb(Z)} + \text{DuréeP} + \text{Acc(Z)} \geq 8$
S AVANT FIN	:	$-\text{Déb(S)} + \text{DuréeP} + \text{Acc(S)} \geq 6$
DURÉE PROJ	:	$\text{DuréeP} \leq 18$
MAXACCEL T	:	$\text{Acc(T)} \leq 3$
MAXACCEL W	:	$\text{Acc(W)} \leq 2$
MAXACCEL X	:	$\text{Acc(X)} \leq 1$
MAXACCEL Y	:	$\text{Acc(Y)} \leq 2$
MAXACCEL Z	:	$\text{Acc(Z)} \leq 3$
MAXACCEL R	:	$\text{Acc(R)} \leq 1$
MAXACCEL S	:	$\text{Acc(S)} \leq 1$

En conclusion

D'un point de vue mathématique, les deux approches – réseau potentiels-étapes ou réseau potentiels-tâches – sont équivalentes et donnent les mêmes résultats. La première est la plus ancienne, la mieux documentée et traditionnellement la plus utilisée en Amérique du Nord. L'interprétation des sommets comme des événements permet au gestionnaire de scinder naturellement le projet en «étapes», de réviser la planification du projet aux événements charnières que seront certains sommets.

Par contre, le tracé d'un réseau potentiels-étapes peut s'avérer difficile lorsque plusieurs tâches admettent des ensembles de prédécesseurs immédiats différents mais non disjoints. Le tracé du réseau potentiels-tâches s'obtient toujours plus ou moins automatiquement. Il n'est d'ailleurs pas indispensable pour calculer moments et marges, car les formules (1)-(5) ne font appel qu'aux relations de prédécesseurs immédiats et aux durées, données qui font partie de la description du projet; les formules correspondantes de l'approche potentiels-étapes reposent de façon cruciale sur la numérotation des sommets et exigent le tracé préalable du graphique. Les réseaux potentiels-tâches s'adaptent plus facilement à des changements dans les données du projet.

De nos jours, la plupart des logiciels offrent les deux approches et le choix de l'une ou l'autre est affaire de goût, d'habitude... ou de contexte culturel !