RAIRO. RECHERCHE OPÉRATIONNELLE

F. STERBOUL
D. WERTHEIMER

Comment construire un graphe PERT minimal

RAIRO. Recherche opérationnelle, tome 15, n° 1 (1981), p. 85-98
<http://www.numdam.org/item?id=RO_1981__15_1_85_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 14, n° 1, février 1980, p. 85 à 98)

COMMENT CONSTRUIRE UN GRAPHE PERT MINIMAL (*)

par F. STERBOUL et D. WERTHEIMER (1)

Résumé. – *Nous donnons un algorithme permettant, pour un problème d'ordonnancement donné, de construire le graphe PERT ayant un nombre minimal de sommets. Cet article contient la preuve de la validité de l'algorithme, un exemple et des résultats numériques.*

Mots clés : ordonnancement, graphe PERT minimal, algorithme de construction de graphes.

Abstract. – *We give an algorithm for constructing, for a given project scheduling problem, the PERT graph having the smallest number of vertices. This paper contains proofs, an example and numerical results.*

Keywords: scheduling, minimal PERT graph, graph construction algorithm.

INTRODUCTION

Les problèmes d'ordonnancement sont définis par la donnée d'un certain nombre d'opérations (les tâches) et des contraintes de succession entre ces tâches (la tâche b ne peut commencer qu'après que la tâche a soit terminée), ainsi que les durées de chacune de ces tâches.

Définir un ordonnancement consiste à attribuer à chacune des tâches une date d'exécution de manière à ce que l'ensemble des tâches soit terminé au plus tôt. Une des méthodes pour résoudre ce problème est la méthode PERT [1, 2]. Le premier pas de cette méthode consiste à construire un graphe (graphe PERT ou américain) dont les arcs représentent les tâches et où les relations de succession entre les tâches sont traduites ainsi : b est successeur de a si et seulement si il existe dans le graphe un chemin dont le premier arc est a et le dernier b. La construction n'est en général possible que si l'on introduit de nouvelles tâches, dites virtuelles, de durée nulle.

Kelley [3] note qu'il est avantageux pour réduire la longueur des calculs suivants de construire un graphe PERT ayant le nombre minimal possible de sommets.

(*) Reçu février 1979.
(1) Université de Lille-I, Villeneuve-d'Ascq, France.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/1981/ 85 /$ 5.00

Le problème ainsi posé par Kelley a été abordé avec plus ou moins de succès par un certain nombre d'auteurs :

Hayes [4] donne un ensemble de recettes pour construire un graphe PERT. Sa méthode ne produit pas, en général, le graphe minimal.

Dimsdale [5] propose un algorithme de construction du graphe PERT minimal.

Fisher, Liebman, Nemhauser [6] montrent que l'algorithme de Dimsdale est faux; ils en donnent un nouveau qui est exact; leur article ne contient pas de preuve mathématique conformément au style du journal (C.A.C.M.).

Cantor et Dimsdale [7] donnent, avec démonstration, un algorithme exact. Leur méthode qui s'applique à un problème qui englobe les problèmes d'ordonnancement, gagne en généralité mais donne lieu, pour notre cas particulier, à des calculs inutilement longs.

Syslo [8] cherche à minimiser le nombre des arcs virtuels ce qui est un autre problème.

Dans le présent article notre contribution est la suivante : comme dans [6] nous ne considérons que le problème d'ordonnancement (ceci implique en particulier que les relations de succession entre tâches réelles ne peuvent pas former de circuit). Nous reprenons certaines notions des articles ci-dessus ou nous en définissons d'autres qui sont voisines. Nous démontrons certaines propriétés (notamment le théorème 1) qui ont échappé aux auteurs précédents et qui permettent finalement de donner un algorithme de construction du graphe PERT minimal réduisant le nombre des calculs.

1. NOTATIONS ET DÉFINITIONS

$G=(X, U)$ graphe orienté dont X est l'ensemble des sommets et U l'ensemble des arcs.

(i, j) arc reliant le sommet i au sommet j.

$u \to v$ indique qu'il existe un chemin dont le premier arc est u et dont le dernier arc est v.

$\forall i \in X$ on note :

$$\mathscr{P}(i)=\{ j \in X : (j, i) \in U \},$$

$$\mathscr{Q}(i)=\{ j \in X : (i, j) \in U \},$$

$$\tilde{\mathscr{P}}(i)=\{ j \in X : \exists \text{ un chemin de } j \text{ vers } i \}.$$

Arc redondant : l'arc (i, j) est dit redondant s'il existe un chemin de longueur supérieure à un de i vers j.

Graphe arc-dual d'un graphe donné

Soit $G=(X, U)$ un graphe donné, *sans circuit* et *sans arc redondant* et soit $H=(Y, V)$ un graphe.

S'il existe une application $f$injective : $X \to V$ telle que $\forall j, \forall i \in X\, i \in \tilde{\mathscr{P}}(j)$ ssi $f(i) \to f(j)$, H sera dit graphe arc-dual de G par l'application f.

REMARQUE 1.1 : 1. H peut être un multigraphe.

2. H ne contient aucun circuit contenant au moins un arc de la forme $f(i)$ car sinon la tâche i se succèderait à elle-même.

Graphe français. [9] *Graphe américain*

Les données du problème d'ordonnancement sont représentées par le graphe « français » $G=(X, U)$, où les sommets représentent les tâches et où l'arc (i, j) appartient à U si et seulement si la tâche i précède immédiatement la tâche j. Le but du problème est donc de construire un graphe H arc-dual de G (H est le graphe PERT ou « américain »); on doit minimiser le nombre des sommets de H.

2. CONSTRUCTION

2.1. Graphe H_0 : construction et propriétés

A partir du graphe français $G=(X, U)$, *sans circuit* et *ne contenant aucun arc redondant* on construit un graphe $H_0=(Y_0, V_0)$ de la manière suivante :

- pour chaque sommet $i \in X$, on définit deux sommets a_i et b_i;
- on pose donc $Y_0 = \bigcup_{i \in X} \{a_i\} \cup \{b_i\}$;
- V_0 est constitué des arcs (a_i, b_i) pour tout $i \in X$, ainsi que des arcs (b_i, a_j) si $i \in \mathscr{P}(j)$ dans G.

Les arcs de la forme (a_i, b_i) seront dits réels et ceux de la forme (b_i, a_j) seront dits virtuels.

Posons $f(i)=(a_i, b_i)$.

PROPOSITION 1 : *H_0 est graphe arc-dual du graphe G par l'application f.*

Il est immédiat que $f(i) \to f(j)$ ssi $i \in \tilde{\mathscr{P}}(j)$ et que f est une application injective.

REMARQUE 2.1.1 : (*a*) H_0 ne contient aucun circuit.

(*b*) H_0 ne contient aucun arc redondant.

(*c*) les différents chemins de H_0 sont formés d'arcs alternativement réels et virtuels.

(*d*) H_0 contient $2|X|$ sommets.

2.2. Graphe H_1 : construction et propriétés

Dans le graphe H_0 on pose :

$$A = \cup \{a_i\}, \qquad B = \cup \{b_i\}$$

et

$$a_i R a_j \text{ ssi } \mathscr{P}(i) = \mathscr{P}(j),$$
$$b_i S b_j \text{ ssi } \mathscr{Q}(i) = \mathscr{Q}(j).$$

On vérifie aisément que les deux relations R et S sont des relations d'équivalence; soient $\bar{a}_1, \ldots, \bar{a}_L, \bar{b}_1, \ldots, \bar{b}_M$ les classes correspondantes sur A et B.

On appelle *contraction de type* 1 l'opération qui consiste dans H_0 à contracter, en un sommet unique, tous les sommets d'une même classe.

Construction du graphe $H_1 = (Y_1, V_1)$

Dans le graphe H_0 on effectue toutes les contractions possibles de type 1 et l'on note $H_1 = (Y_1, V_1)$ le graphe obtenu.

L'application f induit une application de $X \to V_1$ que l'on continue à noter f, pour simplifier les notations. Les sommets de H_1 seront notés $\bar{a}_1, \ldots, \bar{a}_L, \bar{b}_1, \ldots, \bar{b}_M$.

PROPOSITION 2 : *H_1 est graphe arc-dual du graphe G par l'application f.*

On vérifie aisément que les contractions de type 1 n'ont pas altéré les relations de succession et de non-succession du graphe H_0. D'autre part f reste une application injective car si, par exemple, $f(i)$ et $f(j)$ ont, après contraction, même origine et même extrémité, on les considère comme deux arcs différents. Au contraire si des arcs virtuels se trouvent doublés on ne les prend en compte qu'une seule fois.

REMARQUE 2.2.1. – (*a*) H_1 ne contient ni circuit ni arc redondant.

(*b*) Comme dans H_0, les chemins de H_1 sont constitués alternativement d'arcs réels et virtuels.

Définition d'un bon arc de H_1

Un arc $(\bar{b}_i, \bar{a}_j)$ est un bon arc de H_1 ssi :

$$\forall k, l \in X, \quad \begin{cases} k \in \mathscr{P}(j) \\ l \in \mathscr{Q}(i) \end{cases} \Rightarrow \quad k \in \tilde{\mathscr{P}}(l).$$

On appelle *contraction de type* 2 l'opération qui, dans H_1, consiste à contracter, en un sommet unique, le sommet origine et le sommet extrémité d'un bon arc.

REMARQUE 2.2.2 : On vérifie aisément qu'une contraction de type 2 ne supprime aucune relation de succession entre arcs réels qui existait initialement.

REMARQUE 2.2.3 : On vérifie aisément que la définition d'un bon arc $(\bar{b}_i, \bar{a}_j)$ est équivalente à la suivante :

$$\forall k, l \in X, \quad \begin{cases} k \in \tilde{\mathscr{P}}(j) \\ l \in \mathscr{Q}(i) \end{cases} \Rightarrow \quad k \in \tilde{\mathscr{P}}(l).$$

THÉORÈME 1 : *Les bons arcs de H_1 n'ont deux à deux aucun sommet commun.*

(*a*) Montrons que $(\bar{b}_i, \bar{a}_j)$ et $(\bar{b}_i, \bar{a}_l)$ ne peuvent pas être simultanément deux bons arcs de H_1 :

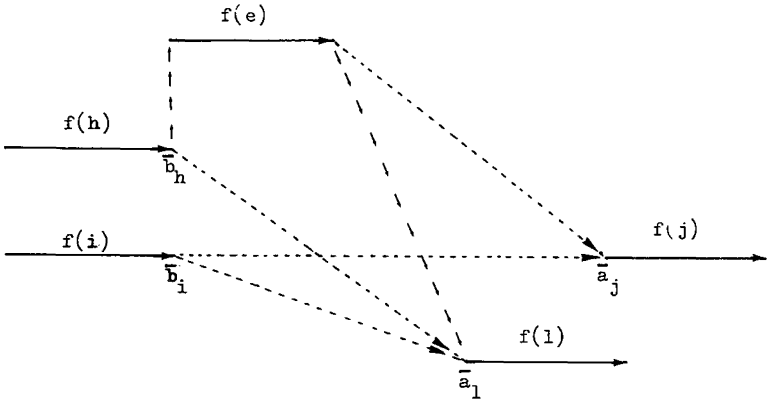


Figure 1

Comme $\bar{a}_j$ et $\bar{a}_l$ sont deux sommets distincts de H_1, on a $\mathscr{P}(j) \neq \mathscr{P}(l)$, donc il existe un sommet h de G tel que, par exemple, $h \in \mathscr{P}(l)$ et $h \notin \mathscr{P}(j)$. Ceci se traduit dans H_1 par l'existence de l'arc $(\bar{b}_h, \bar{a}_l)$, alors qu'il n'existe pas d'arc $(\bar{b}_h, \bar{a}_j)$.

Supposons que $(\bar{b}_i, \bar{a}_l)$ soit un bon arc de H_1. Alors $h \in \mathbf{P}(l)$ et $j \in \mathbf{Q}(i)$ implique $h \in \tilde{\mathbf{P}}(j)$. Compte tenu du fait que $h \notin \mathbf{P}(j)$, il existe alors dans H_1 un chemin dont le premier arc est $f(h)$ et le dernier est $f(j)$ et contenant au moins un autre arc réel. Soit $f(e)$ l'arc de ce chemin tel que $e \in \mathbf{P}(j)$ dans G.

Supposons que l'arc $(\bar{b}_i, \bar{a}_j)$ soit aussi un bon arc de H_1. Alors, $e \in \mathbf{P}(j)$ et $l \in \mathbf{Q}(i)$ implique $e \in \tilde{\mathbf{P}}(l)$. D'où, dans H_1 : $f(e) \to f(l)$. Comme $f(h) \to f(e)$, on a : $f(h) \to f(l)$. Ceci implique que l'arc (h, l) est redondant dans G, contrairement à l'hypothèse.

(*b*) On démontre de même que deux arcs $(\bar{b}_i, \bar{a}_j)$ et $(\bar{b}_h, \bar{a}_j)$ ne peuvent pas être simultanément deux bons arcs de H_1.

2.3. Graphe H_2 : construction et propriétés

Construction du graphe $H_2=(Y_2, V_2)$

Dans le graphe $H_1=(Y_1, V_1)$ on effectue toutes les contractions possibles de type 2.

Soit $H_2=(Y_2, V_2)$ le graphe obtenu.

Dans le but de simplifier les notations :

- on continue à noter les sommets de H_2 comme ceux de H_1;
- on continue à noter f l'application de $X \to V_2$ induite par $f : X \to V_1$.

PROPOSITION 3 : *H_2 est graphe arc-dual de G par l'application f.*

(*a*) Il est immédiat que $f : X \to V_2$ reste une application injective.

(*b*) Montrons qu'il existe un chemin de $f(e)$ vers $f(k)$ si et seulement si $e \in \tilde{\mathscr{P}}(k)$:

(α) $e \in \tilde{\mathscr{P}}(k)$ implique $f(e) \to f(k)$: ceci résulte de la remarque 2.2.2;

(β) $f(e) \to f(k)$ implique $e \in \tilde{\mathscr{P}}(k)$: en effet, $f(e) \to f(k)$ implique qu'il existe dans H_2 un chemin dont le premier arc est $f(e)$ et le dernier arc est $f(k)$. A cause de la transitivité de la relation de succession, on peut supposer que les arcs de ce chemin entre $f(e)$ et $f(k)$ sont tous virtuels (*fig.* 2).

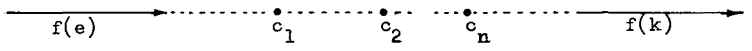


Figure 2

Les sommets intermédiaires $c_1, \ldots, c_m, \ldots, c_n$ proviennent de la contraction des bons arcs $(\bar{b}_{i_1}, \bar{a}_{j_1}), \ldots, (\bar{b}_{i_m}, \bar{a}_{j_m}), \ldots, (\bar{b}_{i_n}, \bar{a}_{j_n})$ de H_1 (*fig.* 3).

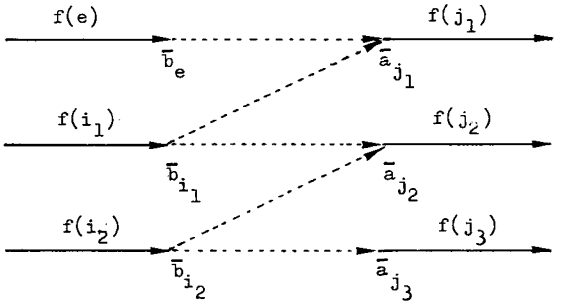


Figure 3

Comme $(\bar{b}_{i_1}, \bar{a}_{j_1})$ est un bon arc, $e \in \mathscr{P}(j_1)$ et $j_2 \in \mathscr{Q}(i_1)$ implique $e \in \tilde{\mathscr{P}}(j_2)$. Comme (b_{i_2}, a_{j_2}) est un bon arc, et d'après la remarque 2.2.3, $e \in \tilde{\mathscr{P}}(j_2)$ et

$j_3 \in \mathscr{Q}(i_2)$ implique $e \in \tilde{\mathscr{P}}(j_3)$. On démontre ainsi de proche en proche que $e \in \tilde{\mathscr{P}}(j_m)$ jusqu'à $m=n$, et finalement $e \in \tilde{\mathscr{P}}(k)$.

Montrons que, parmi les graphes arcs-duaux du graphe G, H_2 a le nombre minimal de sommets. Ce sera une conséquence du théorème suivant.

THÉORÈME 2 : *Soit $H_2=(Y_2, V_2)$ le graphe arc-dual de $G=(X, U)$ par l'application f construit précédemment et soit $H=(Y, V)$ un autre graphe arc-dual de G par l'application g. Alors il existe $Y' \subset Y$ et une application surjective h de Y' sur Y_2 telle que :*

$$\forall i \in X \quad \textit{avec } g(i)=(\alpha, \beta) \quad \textit{alors} \quad f(i)=[h(\alpha), h(\beta)] \quad \textit{avec } \alpha, \beta \in Y'.$$

Démonstration : 1° Construction de h.

Soit $i \in X$ avec $g(i)=(\alpha_i, \beta_i)$ et $f(i)=(\bar{a}_i, \bar{b}_i)$ on pose $h(\alpha_i)=\bar{a}_i$ et $h(\beta_i)=\bar{b}_i$.

2° Montrons que h est bien définie.

Soit $j \in X$, $j \neq i$, avec $g(j)=(\alpha_j, \beta_j)$ et $f(j)=(\bar{a}_j, \bar{b}_j)$; posons $h(\alpha_j)=\bar{a}_j$ et $h(\beta_j)=\bar{b}_j$.

(*a*) Montrons que si $\alpha_j=\alpha_i$ dans H alors $\bar{a}_j=\bar{a}_i$ dans H_2. Il faut donc montrer que $\mathscr{P}(i)=\mathscr{P}(j)$ dans G. Soit $k \in \mathscr{P}(i)$. Comme H est arc-dual de G, on a $g(k) \to g(i)$. Comme $g(i)$ et $g(j)$ ont la même origine, on a $g(k) \to g(j)$, d'où $k \in \tilde{\mathscr{P}}(j)$ dans G. Si on avait $k \notin \mathscr{P}(j)$, il existerait un sommet l de G tel que $l \in \mathscr{P}(j)$ et $k \in \tilde{\mathscr{P}}(l)$ (*fig.* 4).

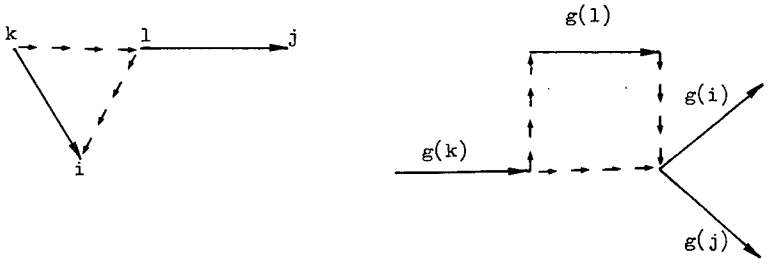


Figure 4

Comme ci-dessus, du fait que $l \in \mathscr{P}(j)$ et que $g(i)$ et $g(j)$ ont la même origine, on a $g(l) \to g(i)$, et $l \in \tilde{\mathscr{P}}(i)$. L'arc (k, i) serait alors redondant dans G, contrairement à l'hypothèse. Donc $k \in \mathscr{P}(j)$, et $\mathscr{P}(i)=\mathscr{P}(j)$.

(*b*) On démontre de même que si $\beta_j=\beta_i$, alors $\bar{b}_j=\bar{b}_i$.

(*c*) Montrons que si $\beta_i=\alpha_j$ dans H alors $\bar{b}_i=\bar{a}_j$ dans H_2. Montrons d'abord que $\beta_i=\alpha_j$ dans H implique que $i \in \mathscr{P}(j)$ dans G. En effet, on a $g(i) \to g(j)$, d'où

$i \in \tilde{\mathscr{P}}(j)$. Si on avait $i \notin \mathscr{P}(j)$, il existerait e tel que $e \in \mathscr{P}(j)$ et $i \in \tilde{\mathscr{P}}(e)$ dans G (*fig.* 5).

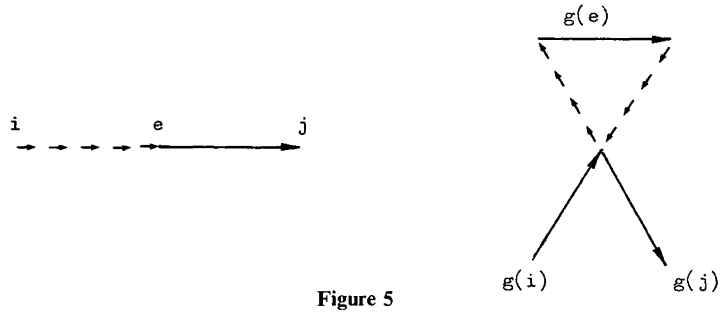


Figure 5

D'où, dans H, $g(i) \to g(e)$ et $g(e) \to g(j)$, ce qui entraînerait l'existence dans H d'un circuit contenant l'arc réel $g(e)$, contrairement à la remarque 1.1. On a donc $i \in \mathscr{P}(j)$, donc $(\bar{b}_i, \bar{a}_j)$ est un arc virtuel de H_1.

Pour montrer que $\bar{b}_i = \bar{a}_j$ dans H_2, il reste donc à montrer que $(\bar{b}_i, \bar{a}_j)$ est un bon arc de H_1. Soit donc $k \in \mathscr{P}(j)$ et $l \in \mathscr{Q}(i)$. Alors, $g(k) \to g(j)$ et $g(i) \to g(l)$. Comme l'origine de $g(j)$ est aussi l'extrémité de $g(i)$, on a $g(k) \to g(l)$, d'où $k \in \tilde{\mathscr{P}}(l)$.

COROLLAIRE : *Le graphe $H_2 = (Y_2, V_2)$ a le nombre minimal de sommets parmi tous les graphes arc-duaux de G. En effet d'après le théorème précédent :*

$$|Y_2| \leqq |Y'| \leqq |Y|.$$

3. ALGORITHME

On décrit maintenant les constructions précédentes sous forme algorithmique. Les sections 1, 2 et 3 correspondent à la construction de H_0 et H_1. La section 4 correspond à la détermination des bons arcs. La section 4.2 est facultative, mais permet une économie d'opérations quand elle vient à être utilisée. L'utilisation de la liste T dans 4.2 et 4.3 permet d'exploiter la propriété démontrée dans le théorème 1. La section 4.5 donne la description finale du graphe H_2 cherché.

On suppose donné le graphe français G, dont les arcs redondants ont été supprimés (par exemple en utilisant les parties I et II de l'algorithme donné dans [6]). Les sommets de G (les tâches) sont représentés par les entiers $i = 1, \ldots, N$. Pour tout i, on connait les listes $\mathscr{P}(i)$, $\mathscr{Q}(i)$ et $\tilde{\mathscr{P}}(i)$.

ALGORITHME : 1. On détermine la partition de $\{1, \ldots, N\}$ en classe $C_1, \ldots, C_L$: i et j appartiennent à la même classe si et seulement si $\mathscr{P}(i) = \mathscr{P}(j)$. On dresse une liste $i_1, \ldots, i_L$ constituée d'un représentant dans chaque classe.

2. On détermine la partition $\{1, \ldots, N\}$ en classes $D_1, \ldots, D_M$: i et j appartiennent à la même classe si et seulement si $\mathscr{Q}(i) = \mathscr{Q}(j)$. On dresse une liste $j_1, \ldots, j_M$ constituée d'un représentant dans chaque classe.

3. Pour tout l, $1 \leqq l \leqq L$, on constitue la liste :

$$V(i_l) = \{ j_m \mid j_m \in \mathscr{P}(i_l),\ 1 \leqq m \leqq M \}$$

et la liste :

$$\tilde{V}(i_l) = \{ j_m \mid j_m \in \mathscr{P}(i_l),\ 1 \leqq m \leqq M \}.$$

Pour tout m, $1 \leqq m \leqq M$, on constitue la liste :

$$W(j_m) = \{ i_l \mid i_l \in \mathscr{Q}(j_m),\ 1 \leqq l \leqq L \}.$$

4. On pose $m = 0$. On utilise une liste T, indexée de 1 à M, vide au départ.

4.1 Augmenter m d'une unité; si $m > M$ aller en 4.5.

Si $W(j_m) = \emptyset$ aller en 4.1.

Si $|W(j_m)| = 1$, soit $\{i_l\} = W(j_m)$, aller en 4.4.

4.2 Soit $E = (\bigcap V(s))_{s \in W(j_m)}$. S'il existe $i_l \in W(j_m)$ tel que $V(i_l) = E$ et i_l ne figure pas dans T, alors aller en 4.4.

4.3 Soit $F = (\bigcap \tilde{V}(s))_{s \in W(j_m)}$. S'il existe $i_l \in W(j_m)$ tel que $V(i_l) \subset F$ et i_l ne figure pas dans T, alors aller en 4.4, sinon aller en 4.1.

4.4 On pose $T(m) = i_l$.

Aller en 4.1.

4.5 L'ensemble des sommets du graphe cherché H_2 est l'ensemble des entiers : $\{2j_1, \ldots, 2j_M\} \cup \{2i_l + 1 \mid i_l$ ne figure pas dans T, $1 \leqq l \leqq L\}$.

L'ensemble des arcs de H_2 est :

Arcs réels : Pour tout $i \in \{1, \ldots, N\}$, soient l et m tels que $i \in C_l$ et $i \in D_m$:

arc$(2i_l + 1, 2j_m)$ si i_l ne figure pas dans T;

arc$(2j_k, 2j_m)$ si $i_l = T(k)$.

Arcs virtuels : Pour tout $m \in \{1, \ldots, M\}$ et tout $i_l \in W(j_m)$:

Arc$(2j_m, 2i_l + 1)$ si i_l ne figure pas dans T;

arc$(2j_m, 2j_k)$ si $i_l = T(k)$, $k \neq m$.

REMARQUE 3.1 : On peut encore diminuer de façon appréciable les calculs, en réduisant, en cours d'algorithme, le nombre des éléments des ensembles $W(j_m)$, $m = 1, \ldots, M$, et $V(i_l)$, $l = 1, \ldots, L$.

Pour cela il suffit de remarquer que, lorsque la section 4.2 est utilisée et qu'il existe $i_l \in W(j_m)$ tel que $V(i_l) = E$, i_l ne figurant pas dans T, alors les arcs (j_k, i_s),

$\forall j_k \in V(i_l) - \{j_m\}$, $\forall i_s \in W(j_m) - \{i_l\}$, deviennent redondants dans la contraction de type 2 relative au bon arc (j_m, i_l). Il est alors possible de retirer :

$$i_s \text{ de } W(j_k) \quad \text{et} \quad j_k \text{ de } V(i_s),$$
$$\forall j_k \in V(i_l) - \{j_m\} \quad \text{et} \quad \forall i_s \in W(j_m) - \{i_l\}.$$

Une légère modification de la section 4.2 de l'algorithme permet de tenir compte de ces résultats.

4. EXEMPLE

Soit le graphe G (*fig.* 6) dont les sommets, au nombre de neuf, sont numérotés de 1 à 9.

i	$P(i)$
1	
2	
3	
4	1, 2, 3
5	1, 3
6	2, 3
7	4, 5
8	4, 5
9	6

i	$Q(i)$
1	4, 5
2	4, 6
3	4, 5, 6
4	7, 8
5	7, 8
6	9
7	
8	
9	

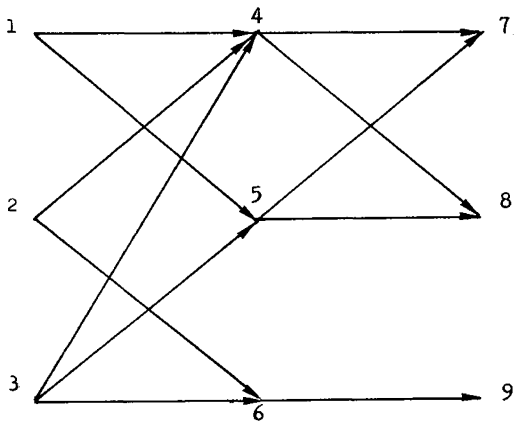


Figure 6

Déroulement de l'algorithme

1. Détermination des classes C_i.

Il y a six classes dont les représentants respectifs constituent la liste i_l, $l = 1, 2, \ldots, 6$.

i_l	1	4	5	6	7	9
l	1	2	3	4	5	6

2. Détermination des classes D_i.

Il y a six classes dont les représentants respectifs constituent la liste j_m, $m = 1, 2, \ldots, 6$.

j_m	1	2	3	4	6	7
m	1	2	3	4	5	6

3. Constitution des tableaux V, W, $\tilde{V}$.

i	$V(i)$	$W(i)$	$\tilde{V}(i)$
1		4, 5	
2	1, 2, 3	4, 6	1, 2, 3
3	1, 3	4, 5, 6	1, 3
4	2, 3	7	2, 3
5	4	9	4, 1, 2, 3
6	6		6, 2, 3

4. Détermination des bons arcs :

— $m=1$, $W(1)=\{4, 5\}$, $E=\{1, 3\}=V(3)$ et $i_3=5$. $(\bar{b}_1, \bar{a}_5)$ est un bon arc et l'on pose $T(1)=5$.

— $m=2$, $W(2)=\{4, 6\}$, $E=\{2, 3\}=V(4)$ et $i_4=6$. $(\bar{b}_2, \bar{a}_6)$ est un bon arc et l'on pose $T(2)=6$.

— $m=3$, $W(3)=\{4, 5, 6\}$. Un seul bon arc est possible $(\bar{b}_3, \bar{a}_4)$ puisque 5 et 6 figurent dans le tableau T, mais $4=i_2$ et $E=\{3\}\neq V(2)$, donc cet arc n'est pas un bon arc.

— $m=4$, $W(4)=\{7\}$ donc $|W(4)|=1$. $(\bar{b}_4, \bar{a}_7)$ est un bon arc et l'on pose $T(4)=7$.

— $m=5$, $W(5)=\{9\}$ donc $|W(5)|=1$. $(\bar{b}_5, \bar{a}_9)$ est un bon arc et l'on pose $T(5)=9$.

— $m=6$, $W(6)=\{\emptyset\}$.

$T(i)$	5	6	0	7	9	0
i	1	2	3	4	5	6

L'ensemble des sommets du graphe H_2 est l'ensemble des entiers :

$$\{2, 4, 6, 8, 12, 14\} \cup \{3, 9\}.$$

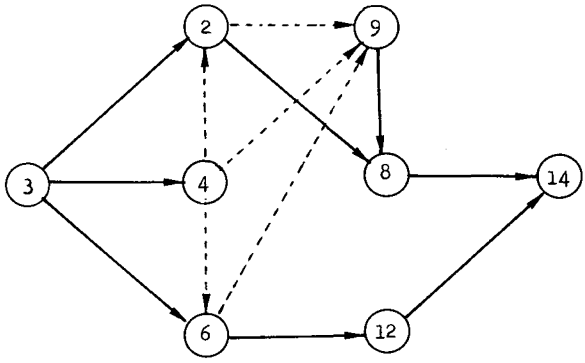
Arcs réels

Arcs	(3, 2)	(3, 6)	(3, 4)	(9, 8)	(2, 8)	(6, 12)	(8, 14)	(8, 14)	(12, 14)
Tâches	1	2	3	4	5	6	7	8	9

Arcs virtuels :

$$(2, 9), (4, 9), (4, 2), (4, 6), (6, 9).$$

Graphe H_2 *:*



5. RÉSULTATS NUMÉRIQUES

L'algorithme a été programmé en Fortran, sur l'Iris 80 du Centre interuniversitaire du Traitement de l'Information de l'Université de Lille-I. Les graphes ont été engendrés aléatoirement. Les temps sont donnés en 1/10 de seconde (*voir* tableau page suivante).

L'examen des résultats permet d'assurer que les temps de calcul sont faibles même pour des graphes dont le nombre de sommets est élevé.

Comme on pouvait le prévoir, pour un nombre de sommets et un nombre d'arcs donnés, le temps de déroulement de l'algorithme dépend du nombre de classes.

Les graphes les mieux adaptés sont les graphes multipartis très denses.

BIBLIOGRAPHIE

1. D. G. MALCOLM, J. H. ROSEBOOM, C. E. CLARK et W. FAZAR, *Application of a Technique for Research and Development Program Evaluation*, Operations Research, vol. 10, n° 6, 1962.
2. C. G. BIGELOW, *Bibliography ou Project Planning and Control by Network Analysis*, Operations Research, vol. 10, n° 5, 1962.
3. J. E. KELLEY, *Critical Path Planning and Scheduling-Mathematical Basis*, Operations Research, vol. 9, n° 3, 1961, p. 296-320.

Nombre de sommets	Nombre d'arcs	Nombre de classes		temps	Nombre de sommets	Nombre d'arcs	Nombre de classes		Temps
		C_i	D_i				C_i	D_i	
30	50	22	18	1	65	60	58	58	4
30	69	25	25	2	65	157	56	55	16
30	160	19	20	6	65	192	55	56	23
30	180	17	15	4	65	233	55	50	16
30	190	10	10	2	65	295	55	54	42
30	200	5	3	1	65	297	50	48	25
35	75	25	27	3	65	378	52	49	43
35	82	29	29	4	65	486	7	7	6
35	112	28	29	7	70	87	58	58	5
35	140	26	25	6	70	144	56	58	12
35	168	24	20	5	70	220	62	64	34
35	196	5	5	2	70	227	61	58	21
40	55	30	29	2	70	251	60	55	24
40	120	31	28	6	70	378	52	43	24
40	144	33	31	9	70	443	58	53	45
40	180	31	31	11	70	506	54	54	64
40	194	31	31	12	70	600	7	7	7
40	300	4	4	2	75	129	63	62	11
45	60	36	37	3	75	169	65	68	23
45	115	35	30	6	75	191	66	65	29
45	144	37	37	11	80	103	68	68	7
45	180	37	36	13	80	210	70	70	34
45	231	38	37	16	80	411	71	68	81
45	270	32	32	16	80	516	68	63	74
45	288	28	24	9	80	605	46	46	35
45	324	5	5	3	85	112	76	75	12
50	93	37	39	5	85	424	77	75	81
50	292	40	37	22	90	118	77	78	10
50	313	39	36	20	90	231	82	85	47
50	400	5	5	4	90	303	88	85	70
55	90	46	44	5	90	380	76	77	69
55	138	48	48	12	90	431	81	80	125
55	180	46	46	20	90	434	79	74	85
55	270	46	45	30	90	493	84	85	149
55	405	6	6	4	90	567	78	74	126
60	74	49	49	4	90	586	78	73	102
60	213	50	46	12	90	636	64	63	87
60	216	55	55	27	90	681	51	51	43
60	265	49	46	19	100	180	88	87	29
60	300	51	49	36	100	269	90	90	72
60	400	49	49	52	100	483	91	92	152
60	450	40	35	24	100	525	91	88	177
60	500	6	6	5	100	900	10	10	14

4. M. HAYES, *The Role of Activity Precedence Relationships in Node-orientated networks*, Project Planning by Network Analysis, North-Holland Publishing Company, Amsterdam, 1969, p. 128.
5. B. DIMSDALE, *Computer Construction of Minimal Project Networks*, I.B.M. Systems J., vol. 2, mars 1963, p. 24-36.
6. A. C. FISHER, J. S. LIEBMAN et G. L. NEMHAUSER, *Computer Construction of Project Networks*, Comm. A.C.M., vol. 11, n° 7, juillet 1968.
7. D. G. CANTOR et B. DIMSDALE, *On Direction-Preserving Maps of Graphs*, J. Comb. Theor., vol. 6, 1969, p. 165-176.
8. M. M. SYSLO, *Optimal Constructions of Reversible Digraph*, preprint.
9. B. ROY, *Graphes et ordonnancement*, Revue française de Recherche opérationnelle, n° 25, 4e trimestre 1962.