Graphe et ordonnancement de projet

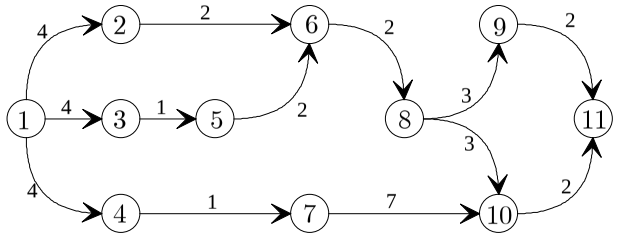
I. Ordonnancement de projet

La réalisation d'un projet (comme la construction d'une maison) passe par l'exécution de différentes tâches.
Certaines tâches peuvent être réalisées simultanément (une équipe peut poser les huisseries tandis qu'une autre s'occupe de la plomberie).
Certaines tâches sont nécessairement antérieures à d'autres (les murs doivent être dressés avant de poser la charpente).
Chaque tâche à une durée qu'on suppose pouvoir estimer[1].
Ordonner un projet consiste à organiser dans le temps chacune de ces tâches avec les objectifs suivants :
+ respecter les conditions d'antériorité,
+ minimiser la durée totale d'exécution.
Par exemple, considérons les tâches numérotées suivantes relatives à la construction d'un entrepôt :

N°	Tâche	Tâche(s) prérequise(s)	durée (en jours)
1	Acceptation des plans		4
2	Creusage fondation	1	2
3	Commande des matériaux	1	1
4	Commande des huisseries	1	1
5	Livraison matériaux	3	2
6	Coulage des fondations	2 et 5	2
7	Livraisons des huisseries	4	7
8	Pose des murs	6	3
9	Pose du toit	8	2
10	Pose des huisseries	7 et 8	2
11	Livraison de l'entrepôt	9 et 10	

La *méthode P.E.R.T*[2] consiste à mettre en ordre sous forme d'un réseau chacune des tâches précédentes en visualisant les conditions d'antériorité et les durées d'exécution des tâches. Pour cela, on représente chaque tâche sur une figure et on visualise la nécessité de réalisation de la tâche i avant l'exécution d'une tâche j par un flèche allant de i à j, flèche où nous ferons figurer la durée de la tâche i.

Le projet ci-dessus peut alors se visualiser sous la forme suivante :



A l'aide d'un tel graphe, on peut ordonner « à la main » le projet, c'est à dire prévoir la date de début au plus tôt de chaque tâche. On peut aussi déterminer la durée minimale d'exécution du projet, celle-ci correspondant à la longueur du plus long chemin menant, la tâche d'acceptation des plans, à la tâche de livraison. Dans l'exemple ci-dessus, cette longueur est de 14 jours et correspond à l'exécution des tâches 1, 4, 7, 10. Le chemin correspondant est appelé chemin critique, tout retard pris le long de ce chemin rallonge d'autant la durée

[1] : Pour estimer la durée D d'une tâche, il est usuel de procéder ainsi :
+ on évalue la durée la plus optimiste notée O,
+ on évalue la durée la plus pessimiste P,
+ on évalue la durée la plus réaliste R.

On définit alors D par le calcul barycentrique : $D = \frac{O + 4 \times R + P}{6}$.

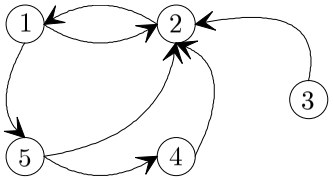
[2] : P.E.R.T. pour « Programm Evaluation and Review Technic » ou « technique d'ordonnancement et de contrôle des programmes ». Inventée en 1957 par l'US Navy pour le projet POLARIS, cette technique a permis de coordonner les travaux de près de 6000 constructeurs dans les délais imposés par le gouvernement américain. Ce projet POLARIS représentait entre autres : 250 fournisseurs, 9000 sous-traitants et 7 ans de réalisation. L'utilisation du P.E.R.T. a permis de ramener la durée globale de réalisation du projet de 7 à 4 ans.

d'exécution du projet. En revanche, des retards pris sur les autres tâches n'auront pas d'incidences aussi immédiates, ceci peut permettre quelques battements…

Dans le présent article, nous allons conceptualiser mathématiquement le problème ci-dessus en étudiant d'abord les propriétés des graphes du type précédent puis en voyant ensuite comment on peut ordonnancer ceux-ci de manière systématique.

II. Définitions générales

Un *graphe orienté* est constitué d'un ensemble d'éléments, appelés *sommets*, qui peuvent être reliés entre eux par des flèches, appelées *arcs*. Un graphe peut se visualiser sous la forme d'un *diagramme sagittal* comme ci-dessous (où les sommets sont numérotés de 1 à 5) :



Le vocabulaire relatif aux graphes, n'étant pas entièrement standardisé, convenons d'adopter le suivant :

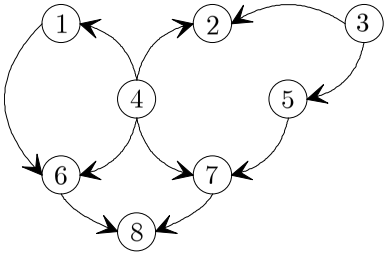
Lorsqu'un arc va du sommet i au sommet j, on dit que i est *l'extrémité initiale* et j *l'extrémité finale* de l'arc. Etant donnés deux sommets i et j, s'il existe un arc allant de i à j, on dit que i est un *prédécesseur* de j et qu'inversement j est un *successeur* de i.

On appelle *chemin*, toute succession d'arcs telle que l'extrémité finale de chaque arc, correspond à l'extrémité initiale du suivant. Dans l'exemple ci-dessus, il existe un chemin allant de 3 à 5, chemin obtenu en transitant par les sommets 2 et 1.

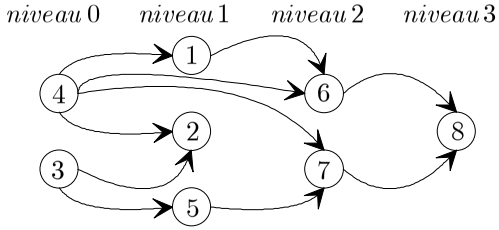
Enfin on appelle *circuit*, tout chemin formé d'arcs distincts, se refermant sur lui-même. Dans l'exemple ci-dessus, on peut former un circuit liant les sommets 1,5,4,2 dans cet ordre.

III. Hiérarchisation d'un graphe sans circuits

Un graphe sans circuits peut être hiérarchisé par niveaux de sorte que tout arc du graphe mène d'un sommet vers un sommet de niveau supérieur. Par exemple, le graphe :



peut être hiérarchisé en :



Pour hiérarchiser de manière systématique un graphe sans circuits, on peut exploiter l'algorithme suivant :

Initialisation
    Affecter tous les sommets du niveau $+\infty$.
    Poser $k = 0$.

Déroulement de la hiérarchisation
    Tant qu'il reste des sommets de niveau $+\infty$:
        Pour chaque sommet j de niveau $+\infty$:
            Si tous les prédécesseurs du sommet j sont de niveau $< k$
            Alors poser le niveau du sommet j égal à k.
            Fin Si.
        Fin Pour.
        Incrémenter k.
    Fin Tant que.

Fin.

Cet algorithme commence par affecter du niveau 0 tous les sommets qui n'ont aucun prédécesseur. Puis, une fois les niveaux 0,1,…,$k-1$ déterminés, l'algorithme cherche les sommets dont tous les prédécesseurs sont de niveau $< k$ pour leur attribuer le niveau k.

Pour le graphe précédent, l'évolution du niveau des sommets lors des passages dans la boucle *Tant que* est donnée par le tableau suivant :

Sommets	$k=0$	$k=1$	$k=2$	$k=3$
1	$+\infty$	1	1	1
2	$+\infty$	1	1	1
3	0	0	0	0
4	0	0	0	0
5	$+\infty$	1	1	1
6	$+\infty$	$+\infty$	2	2
7	$+\infty$	$+\infty$	2	2
8	$+\infty$	$+\infty$	$+\infty$	3

Enfin notons que si cet algorithme est appliqué à un graphe présentant un circuit, celui-ci ne s'arrêtera pas. Cette idée peut d'ailleurs être exploitée pour modifier l'algorithme précédent de sorte de tester si un graphe possède un circuit ou non.

IV. Diagramme d'ordonnancement de projet

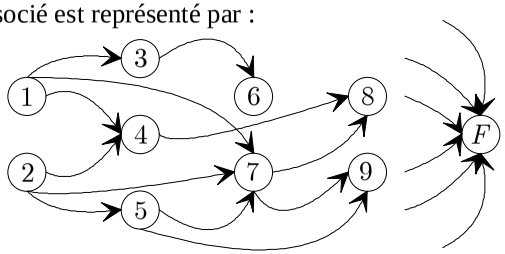
Revenons maintenant au cadre initiale de notre étude.
Considérons un projet constitué de différentes tâches, certaines étant préalables à d'autres. A partir de ce projet nous formons un graphe orienté :
+ les sommets correspondent aux tâches,
+ la nécessité de réaliser une tâche i avant une tâche j est figurée par un arc allant du sommet i au sommet j.

A ce graphe, nous ajoutons un sommet dit de *finalisation*, noté F, et nous relions tous les autres sommets par un arc menant à celui-ci.
La possibilité d'achèvement du projet se traduit par l'inexistence de circuits à l'intérieur de ce graphe. Cela permet de hiérarchiser ce dernier, le sommet de finalisation se trouvant au niveau le plus élevé.

Concrètement, supposons disposer d'un projet formé de 9 tâches dont les interdépendances sont données par le tableau ci-dessous :

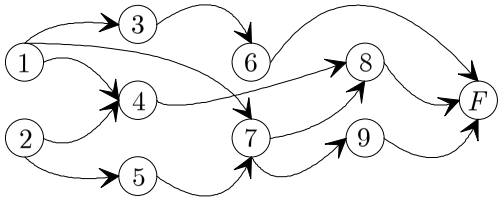
Tâche	Tâches préalables
1	aucune
2	aucune
3	1
4	2,1
5	2
6	3
7	1,2,5
8	4,7
9	5,7

Le graphe orienté associé est représenté par :



(les tâches ont été numérotées par niveaux croissants, par commodité, et les flèches vers le sommet de finalisation n'ont pas été représentées, pour la lisibilité…).

La réalisation du projet passant par la réalisation de *toutes* les tâches, certains arcs sont inutiles. Par exemple, l'arc de 2 à 7 est inutile car 7 nécessite 5 qui nécessite elle-même 2. Par ce principe, on parvient à épurer le graphe précédent et on parvient au graphe suivant, plus léger, où l'on voit quelles sont les véritables dépendances entre les différentes tâches :



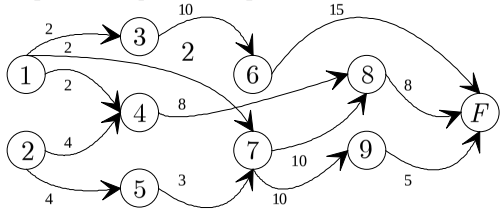
Supposons maintenant que chacune de nos tâches ait une durée prévisible spécifiée par le tableau suivant :

Tâche	Durée
1	2
2	4
3	10

4	8
5	3
6	15
7	10
8	8
9	5

Posons alors les deux problèmes suivants :
(1) Quelle est la meilleure date de commencement de la tâche i ?
(2) Quelle est la plus courte durée d'exécution du projet ?

Pour visualiser sur notre graphe la durée de chaque tâche, on affecte chaque arc au départ de i d'un poids égal à la durée d'exécution de la tâche i. On obtient alors un graphe dit *pondéré* que l'on représente sous la forme suivante :



Pour un graphe pondéré, on appelle *longueur* d'un chemin allant de i à j, la somme des poids des arcs ayant constitué ce chemin.
Voyons maintenant comment se posent les problèmes précédents :

(1) Les tâches de niveau 0 peuvent être commencées immédiatement. Pour tout autre tâche i, la meilleure date de commencement est égale à la durée du *plus long chemin* menant un sommet de niveau 0 au sommet i.

(2) La plus courte durée d'exécution du projet correspond à la date de commencement du sommet de finalisation.

V. Détermination du plus long chemin

Considérons un graphe orienté sans circuits. Nous pouvons hiérarchiser celui-ci par niveaux. Supposons ensuite que notre graphe soit pondéré et notons $poids(i,j)$ le poids de l'arc allant de i à j.

On veut déterminer, pour chaque sommet i, la longueur du plus long chemin menant un sommet de niveau 0 au sommet i. Cette longueur sera notée $L(i)$. On peut l'obtenir par l'algorithme suivant :

Initialisation

Hiérarchiser le graphe par niveaux.
Pour tous les sommets j du graphe
Si le sommet j est de niveau 0
Alors poser $L(j) = 0$
Sinon poser $L(j) = -\infty$
Fin Pour.

Détermination des plus grandes longueurs

Pour chaque niveau k par ordre croissant :
Pour tout sommet i de niveau k :
Pour tout successeur j de i :
Poser $d = L(i) + poids(i,j)$.
Si $d > L(j)$
Alors poser $L(j) = d$.
Fin Si.
Fin Pour
Fin Pour.
Fin Pour.

Fin.

Cet algorithme détermine les $L(i)$ en commençant par les sommets de niveau 0, puis 1,... Lorsque les $L(i)$ des sommets de niveau k sont déterminés, l'algorithme détermine pour chaque successeur j de i, la plus grande longueur menant à j en transitant par i, c'est le rôle de la variable d. Cette longueur est ensuite comparée à la plus grande longueur menant à j déjà déterminée de sorte de conserver la plus grandes des deux.
Revenons à notre projet précédent. Par l'algorithme ci-dessus on obtient :

Tâche	Date de début
1	0
2	0

3	2
4	4
5	4
6	12
7	7
8	17
9	17
F	27

La durée d'exécution du projet est 27.

VI. Généralisations

L'algorithme précédent peut s'adapter à la détermination du plus long chemin allant d'un sommet i à tout autre : il suffit pour cela, lors de l'initialisation de poser $L(j) = -\infty$ pour tout $j \neq i$. Cela a pour effet de supprimer « virtuellement » les sommets $j \neq i$ de niveaux inférieurs ou égaux à i.
Cet algorithme peut encore être adapté à la recherche de plus courts chemins. : il suffit pour cela de changer $-\infty$ en $+\infty$, et d'inverser la comparaison $d > L(j)$ en $d < L(j)$.

Enfin ces idées peuvent s'appliquer à tout graphe orienté sans circuits en considérant tous les arcs affectés du poids 1. On peut déterminer alors s'il existe un chemin menant d'un sommet à un autre, ainsi que la longueur du plus long/court de ces chemins.