

THÉORIE DES GRAPHS

PROGRAMME:

- ❖ **Concepts fondamentaux de la théorie des graphes:**
 - Définitions
 - Structure d'un graphe
 - Graphes particuliers
 - Modes de représentations des graphes
- ❖ **Connexité dans un graphe:**
 - Cheminements dans un graphe
 - Connexité
 - Forte connexité
 - La mise en ordre d'un graphe connexe ou la recherche d'un circuit.
- ❖ **Arbres et arborescences:**
 - Arbres et arborescences
 - Le problème de recherche d'un arbre de poids minimum
 - Le problème de recherche d'un plus court chemin
- ❖ **Problème du flot maximum:**
 - Définitions
 - Le problème de recherche du flot maximum.
- ❖ **Problème d'ordonnancement:**
 - La représentation du réseau PERT
 - La détermination du calendrier des dates au plus tôt et des dates plus tard
 - Analyse et identification des tâches critiques.
- ❖ **Cheminements remarquables:**
 - Les cheminements eulériens
 - Les chemins hamiltoniens

UN BREF HISTORIQUE DE LA THÉORIE DES GRAPHS

Tout le monde s'accorde à considérer que la théorie des graphes est née en 1736 avec la communication d'Euler (1707-1783) dans laquelle il proposait une solution au célèbre problème des ponts de Königsberg. Le problème posé était le suivant. Deux îles A et D sur la rivière Pregel à Königsberg étaient reliées entre elles ainsi qu'aux rivages B et C à l'aide de sept ponts (désignés par des lettres minuscules) comme le montre la figure 1.

Le problème posé consistait, à partir d'une terre quelconque A, B, C, ou D, à traverser chacun des ponts une fois et une seule et à revenir à son point de départ. Euler représenta cette situation à l'aide d'un "dessin" où les sommets représentent les terres et les arêtes, les ponts comme le montre la figure 2.

Le problème des ponts de Königsberg est identique à celui consistant à tracer une figure géométrique sans lever le crayon et sans repasser plusieurs fois sur un même trait.

La représentation d'un problème par un dessin contribue souvent à sa compréhension (Pb nettoyage),

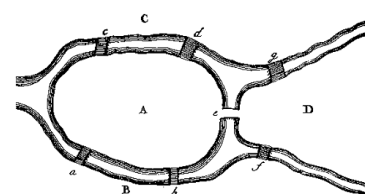


FIG. 1 – La rivière Pregel et l'île de Kneiphof

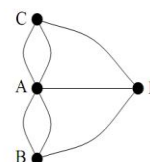


FIG. 2 – Graphe associé au problème des ponts de Königsberg

CHAPITRE 1: CONCEPTS FONDAMENTAUX DE LA THÉORIE DES GRAPHS

1- Définitions:

1-1 Qu'est ce qu'un graphe?

C'est en 1822 que le mot "graphe" est introduit par l'Anglais J.J.Sylvester, et en 1936 que parait le premier livre sur la théorie des graphes, écrit par D.King.

Un graphe est un dessin géométrique défini par la donnée d'un ensemble de points (appelés sommets ou nœuds), reliés entre eux par un ensemble de lignes ou de flèches (appelées arêtes ou arcs). Chaque arête a pour extrémités deux points, éventuellement confondus.

Les graphes peuvent servir à représenter un grand nombre de situations courantes comme:

- Les liens routiers
- Les réseaux de communication
- Les circuits électriques
- Les liens entre diverses personnes ou entités administratives.

Exemple: La figure suivante représente un plan de circulation à sens unique d'une ville où chaque localité est représentée par un point appelé sommet et chaque route par un arc orienté indiquant le sens de la circulation.

Ainsi les notions qu'on peut définir sur un graphe, vont servir à résoudre certains problèmes liés à différents domaines.

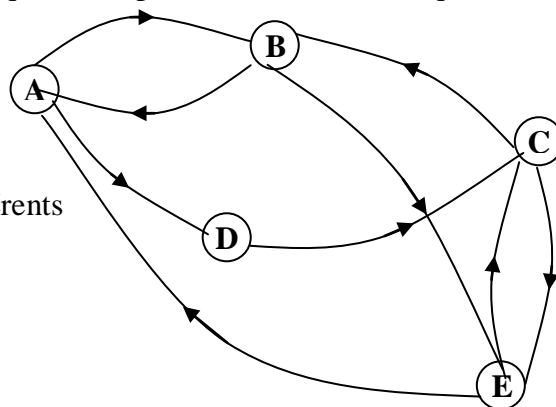


Figure 1

1-2 Graphe orienté:

Un graphe orienté est un système formé d'un ensemble fini de sommets que l'on notera $\{x_1, x_2, \dots, x_n\}$ et d'un ensemble fini d'arcs reliant dans un ordre bien défini ces sommets, ou un certain nombre d'entre eux noté $\{u_1, u_2, \dots, u_m\}$.

Exemple:

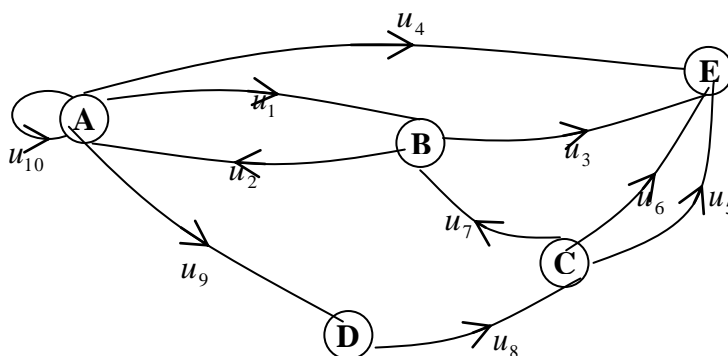


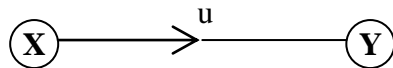
Figure 2

Mathématiquement, un graphe orienté est représenté par le couple $G(X, U)$, où:

- X est l'ensemble des sommets.
- U est l'ensemble des arcs.

❖ Notation:

On note un arc reliant un sommet x au sommet y dans un graphe G : par $u=(x,y)$.



Si le graphe G contient n sommets, on dit alors que G d'ordre n .

Chaque arc du graphe G relie respectivement deux sommets, le sommet de départ qui représente l'extrémité initiale de l'arc et le sommet d'arrivée qui représente l'extrémité terminale.

❖ Autrement dit:

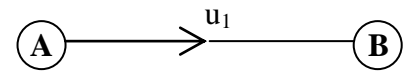
Un graphe orienté est défini par le quadruplet: $G=(X,U,I,T)$ où

- I est l'application extrémité initiale d'un arc définie par:
 $I: U \rightarrow X$
 $(x,y) \rightarrow I(x,y)=x$
- T est l'application extrémité terminale d'un arc définie par:
 $T: U \rightarrow X$
 $(x,y) \rightarrow T(x,y)=y$

Exemple:

Soit $u_1=(A,B)$ un arc de l'ensemble des arcs U du graphe G ci-dessus:

$I(u_1)=A$ et $T(u_1)=B$



Remarque:

On appelle l'arc dont l'extrémité initiale est confondue avec l'extrémité terminale une boucle notée $u(x,x)$

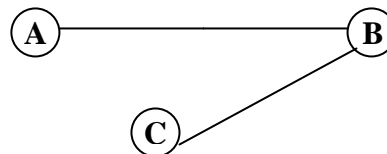
Exemple:

Dans la figure 3 $u_{10}=(A,A)$ / $I(u_{10})=A$ et $T(u_{10})=A$. L'arc u_{10} est une boucle.

1-3 Graphe non orienté:

Si on définit une relation sur un ensemble où la notion d'ordre n'est pas importante, on représente ainsi la relation entre sommets par un arc non orienté appelé arête. On obtient alors un graphe non orienté, noté $G=(X,E)$.

Exemple:



Remarque:

Une arête peut être transformée en deux arcs de sens différents



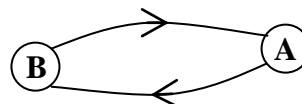
1-4 graphe simple et graphe multiple:

Un graphe simple est un graphe sans boucles ni arcs (arête) multiples. Dans le cas contraire, on dira que le graphe est multiple.

Exemple:



Arêtes multiples



Arcs multiples

On définit ainsi, la multiplicité d'un graphe orienté multiple par le nombre maximum d'arcs ayant la même extrémité initiale et la même extrémité terminale. Soit p ce nombre, on dit G est un p -graphe.

1-5 L'ensemble des prédécesseurs, successeurs et voisins d'un sommet:

Considérons le graphe correspondant à la Figure2:

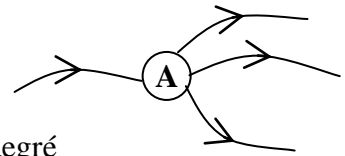
- De B et E on peut atteindre A par BA et EA. Donc, B et E forment l'ensemble des prédécesseurs de A, qu'on note $\Gamma^-(A)$.
- De A on peut atteindre B et D par AB et AD. Donc, B et D forment l'ensemble des successeurs de A, qu'on note $\Gamma^+(A)$.
- L'ensemble des voisins du sommet A est égale à la réunion de l'ensemble de ses prédécesseurs et de ses successeurs.

L'application Γ qui, à tout élément de X, fait correspondre une partie de X est appelée une application multivoque.

1-6 Degré d'un sommet:

Soit le graphe de la figure 3, considérons le sommet A:

- A est l'extrémité initiale de 3 arcs, on dit alors que le demi-degré extérieur de A est 3, on le note $d_G^+(A) = 3$.
- A est l'extrémité terminale d'un seul arc, on dit alors que le demi-degré intérieur de A est 1, on le note $d_G^-(A) = 1$.
- La somme du demi-degré intérieur et du demi-degré extérieur du A définit le degré de A, on le note $d(A) = 4$.



Remarque:

Degré de A égale à 0 \rightarrow sommet isolé.

Degré de A égale à 1 \rightarrow sommet pendant.

2- Structure d'un graphe:

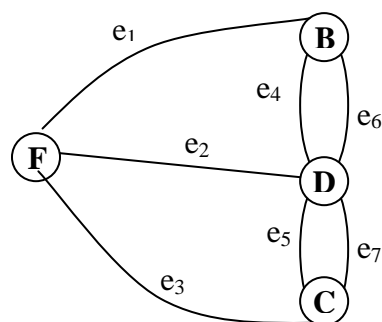
Considérons le réseau routier de l'Algérie $G=(X,U)$ tel que:

X représente l'ensemble des villes d'Algérie et U représente l'ensemble des routes nationales et départementales algériennes.

- Soit $A \subset X$, l'ensemble des villes de la wilaya de Tizi Ouzou et U_A l'ensemble des routes reliant ces villes. On définit ainsi le graphe $G_A=(A,U_A)$ dit sous-graphe de G, représentant l'ensemble du réseau routier de la wilaya de Tizi Ouezou.
- Soit $W \subset U$, l'ensemble des routes départementales Algériennes. On définit ainsi le graphe $G_W=(X,W)$, dit graphe partiel de G représentant les routes départementales Algériennes.
- Soient U_A l'ensemble des routes reliant les villes de la wilaya de Tizi Ouezou (nationales et départementales) et W l'ensemble des routes départementales Algériennes. On définit ainsi le graphe $G_{AW}=(A,W \cap U_A)$, dit sous-graphe partiel de G représentant l'ensemble des routes départementales de la wilaya de Tizi Ouezou.

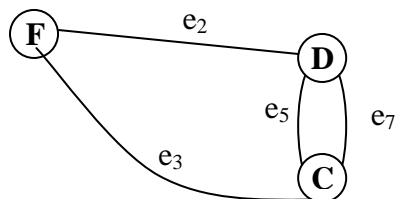
Exemple:

Soit le graphe $G=(X,U)$:

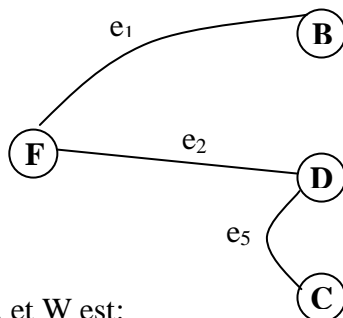


Soient $A = \{F, D, C\}$ et $W = \{e_1, e_2, e_5\}$

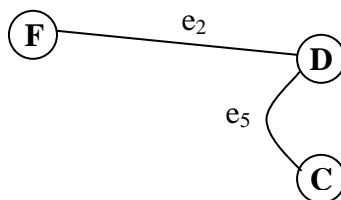
- a) Le sous-graphe engendré par A est le graphe $G_A = (A, E_A)$, avec $E_A = \{e_2, e_3, e_5, e_7\}$



- b) Le graphe partiel engendré par W est:



- c) Le sous-graphe engendré par A et W est:

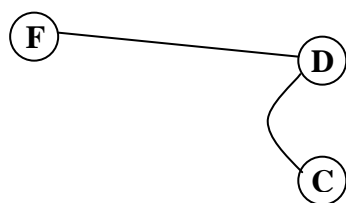


3-Les graphes particuliers:

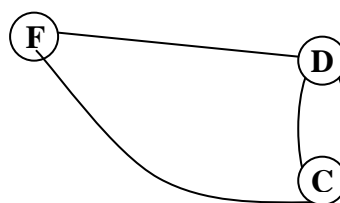
3-1 Graphe complet:

On appelle graphe complet un graphe dont tous les sommets sont adjacents (si, pour toute paire de sommets, il existe au moins un arc).

Exemple:



G_1



G_2

Les sommets F et C dans G_1 ne sont pas adjacents, le graphe est donc non complet.

Les sommets du graphe G_2 sont tous adjacents, d'où le graphe G_2 est complet.

Si un graphe G est simple et complet, d'ordre n , on le note K_n (Ex: G_2 est noté par K_3).

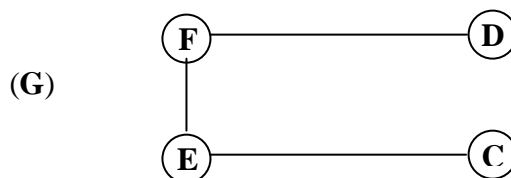
3-2 Graphe complémentaire:

A un graphe simple $G=(X,U)$, on peut définir un graphe complémentaire $\bar{G}=(X,\bar{U})$ comme suit: $u \in \bar{U} \Leftrightarrow u \notin U$.

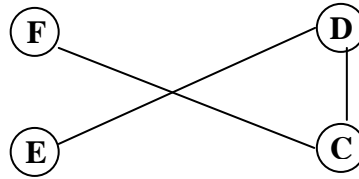
C'est-à-dire: une arête (arc) appartient au graphe complémentaire (\bar{G}) si elle n'appartient pas au graphe initial G .

Exemple:

On considère le graphe simple suivant:



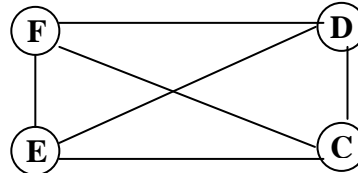
Son graphe complémentaire (\bar{G}) est:



Conséquence:

$G \cup \bar{G}$ est un graphe simple complet, donc un K_n .

$$G \cup \bar{G} = K_4$$



3-3 Graphe planaire:

Un graphe est dit planaire si on peut le dessiner sur un plan de telle façon que les arêtes ne se coupent pas, en dehors de leur extrémités.

3-4 Graphe biparti:

Un graphe est biparti si l'ensemble de ses sommets peut être réparti en deux classes X_1 et X_2 telles que, deux sommets de la même classe ne soient pas adjacents.

4- Modes de représentation d'un graphe

4-1 La représentation matricielle:

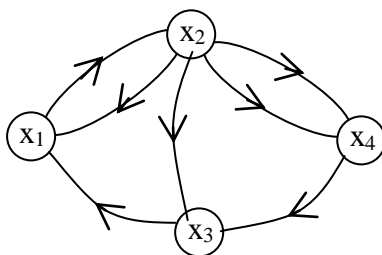
A un graphe $G=(X,U)$ contenant n sommets et m arcs, on associera trois types de matrices:

4-1-1 La matrice d'adjacence:

La matrice d'adjacence du graphe $G=(X,U)$ est une matrice $n*n$, ses éléments prennent deux valeurs 1 ou 0. Chaque ligne et chaque colonne correspondent à un sommet du graphe. Ainsi chaque élément de la matrice indique la relation qui existe entre deux sommets:

- 1 signifie que les deux sommets sont reliés par un arc orienté.
- 0 signifie que les deux sommets ne sont pas reliés par un arc.

Exemple:



(G)

La matrice d'adjacence de G est la suivante:

	x_1	x_2	x_3	x_4
x_1	0	1	0	0
x_2	1	0	1	1
x_3	1	0	0	0
x_4	0	0	1	0

- $u=(x_1,x_2)$ est arc du graphe $G \rightarrow a_{12}=1$.
- pas d'arc ayant comme extrémité initiale x_1 et extrémité terminale $x_3 \rightarrow a_{13}=0$.

Notation:

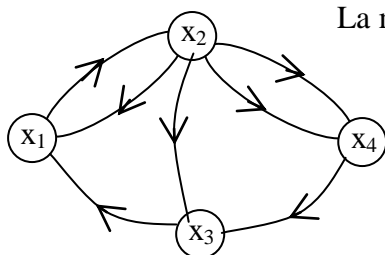
Les éléments de la matrice d'adjacence sont définis par:

$$a_{ij} = \begin{cases} 1 & \text{s'il existe un arc orienté } (x_i, x_j) \\ 0 & \text{sinon} \end{cases}$$

4-1-2 La matrice associée:

La matrice associée du graphe $G=(X,U)$ est une matrice $n*n$, où chaque ligne et chaque colonne correspondent à un sommet du graphe, les éléments de la matrice associée indique le nombre d'arcs orientés dans le même sens reliant deux sommets.

Exemple:



(G)

La matrice associée de G est la suivante:

	x_1	x_2	x_3	x_4
x_1	0	1	0	0
x_2	1	0	1	2
x_3	1	0	0	0
x_4	0	0	1	0

Notation:

Les éléments de la matrice associée sont définis par: $a_{ij}=k$ si on a k arcs orientés de la forme (x_i, x_j) .

Remarque:

Dans la matrice associée on a:

- La somme des valeurs d'une ligne détermine le demi-degré extérieur du sommet correspondant.
- La somme des valeurs d'une colonne détermine le demi-degré intérieur du sommet correspondant

4-1-3 La matrice d'incidence aux arcs:

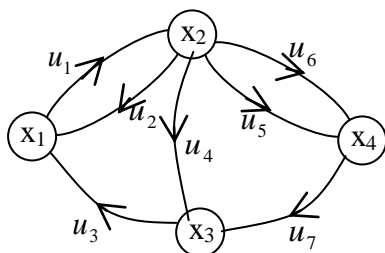
La matrice d'incidence aux arcs d'un graphe $G=(X,U)$ est une matrice $n*m$, ses éléments prennent les valeurs 1, 0 ou -1. Chaque ligne de la matrice est associée à un sommet et chaque colonne à un arc. Chaque élément de la matrice indique la relation entre un sommet et un arc comme suit:

- +1 signifie que le sommet est une extrémité initiale de l'arc.
- -1 signifie que le sommet est une extrémité terminale de l'arc.
- 0 signifie qu'il n'existe pas de relations entre le sommet et l'arc.

Exemple:

Soit un graphe d'ordre 4, composé de 7 arcs.

La matrice d'incidence aux arcs de G est:



(G)

	u_1	u_2	u_3	u_4	u_5	u_6	u_7
x_1	+1	-1	-1	0	0	0	0
x_2	-1	+1	0	+1	+1	+1	0
x_3	0	0	+1	-1	0	0	-1
x_4	0	0	0	0	-1	-1	+1

Remarque: Cette matrice ne convient pas pour les graphes avec boucles.

Notation:

Les éléments de la matrice d'incidence aux arcs sont définis par:

$$a_{ij} = \begin{cases} +1 & \text{si } x_i \text{ est l'extrémité initiale de l'arc } u_j \\ -1 & \text{si } x_i \text{ est l'extrémité terminale de l'arc } u_j \\ 0 & \text{dans les autres cas.} \end{cases}$$

Remarque:

Dans la matrice d'incidence on a:

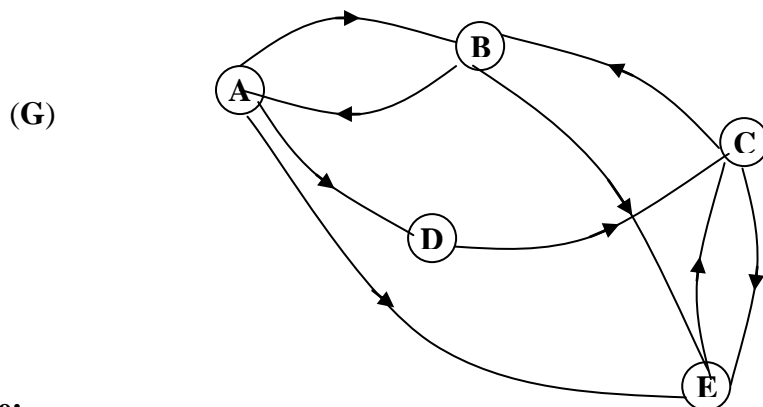
- Le nombre de valeurs égale à +1 d'une ligne donne le degré extérieur du sommet correspondant.
- Le nombre de valeurs égale à -1 d'une ligne donne le degré intérieur du sommet correspondant.

CHAPITRE 2: CONNEXITÉ DANS UN GRAPHE

1- Cheminements dans un graphe:

Les cheminements dans la théorie des graphes sont de quatre types: la chaîne, le cycle, le chemin et le circuit.

On définira ces notions dans le graphe G:



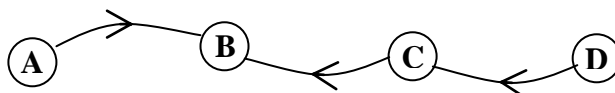
1-1 La chaîne:

Soit $G=(X,U)$ un graphe.

Une chaîne joignant deux sommets x_0 et x_k dans un graphe G est une suite de sommets reliés par des arêtes tels que, deux sommets successifs ont une arête commune. On la note: (x_0, x_1, \dots, x_k) .

Exemple:

Dans le graphe G , (A, B, C, D) est une chaîne.



Une chaîne est dite simple si on passe une seule fois par ses arcs.

Exemple:

Dans le graphe G , (A, B, E) , (A, E, B, A, E) sont des chaînes joignant les sommets A et E , la seconde n'est pas simple car l'arc (AE) est parcouru deux fois.

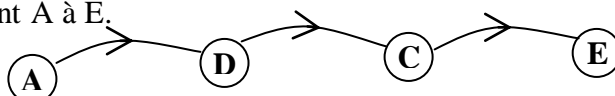
1-2 Le chemin:

Soit $G=(X,U)$ un graphe.

Un chemin du sommet x_0 à x_k dans un graphe G , est une suite de sommets reliés successivement par des arcs orientés dans le même sens. On le note: (x_0, x_1, \dots, x_k) .

Exemple:

Dans le graphe G , (A, D, C, E) est un chemin joignant A à E .



Un chemin est dit simple si on passe une seule fois par ses arcs.

Exemple:

Dans le graphe G , (A, D, C, E) , (A, D, C, E, C, E) sont des chemins joignant les sommets A et E , le second chemin n'est pas simple car l'arc (CE) est parcouru deux fois.

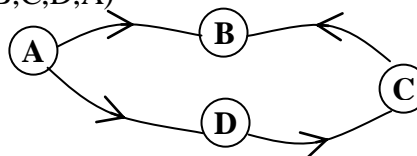
1-3 Le cycle:

Un cycle est une chaîne simple dont les deux extrémités coïncident. On le note $(x_0, x_1, \dots, x_k = x_0)$.

Exemple:

Dans le graphe G , la suite de sommets suivante (A, B, C, D, A)

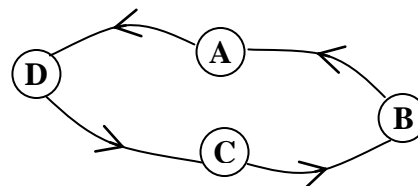
Remarque: une boucle est un cycle particulier.



1-3 Le circuit:

Un circuit est un chemin dont les deux extrémités sont confondues; on le note $(x_0, x_1, \dots, x_k = x_0)$.

Dans le graphe G, la suite (A,D,C,B,A) est un circuit.



Une chaîne (cycle-chemin-circuit) est dite élémentaire si on passe une seule fois par ses sommets (tous les sommets sont différents).

Un chemin eulérien est un chemin simple qui passe une et une seule fois par chaque arc du graphe.

Un chemin hamiltonien est un chemin qui passe une et une seule fois par chaque sommet du graphe.

2- la connexité:

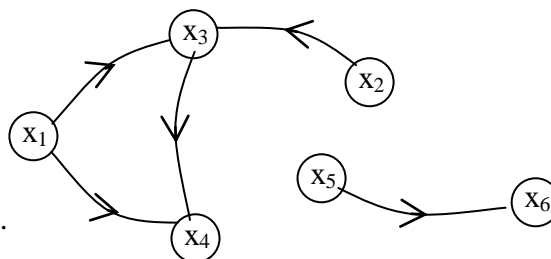
2-1 La notion de connexité:

On définit la connexité dans un graphe, par la relation entre deux sommets de la manière suivante: deux sommets x et y ont une relation de connexité \Leftrightarrow il existe une chaîne entre x et y ou bien $x=y$.

Exemple:

Soit le graphe (G) suivant:

- il existe une chaîne entre le sommet x_1 et x_2 notée $C=(x_1, x_3, x_2)$. Alors x_1 et x_2 ont une relation de connexité.
- il n'existe pas de chaînes entre le sommet x_1 et x_5 , alors x_1 et x_5 n'ont pas une relation de connexité.



2-2 Les composantes connexes:

On appelle composante connexe un ensemble de sommets, qui ont deux à deux la relation de connexité, de plus tout sommet en dehors de la composante n'a pas de relation de connexité avec les sommets de cette composante.

Exemple:

Dans le graphe G de la figure précédente:

- les sommets x_1, x_2, x_3, x_4 ont deux à deux la relation de connexité, donc l'ensemble $\{x_1, x_2, x_3, x_4\}$ forme ainsi la 1^{ère} composante connexe, on la note C_1 .
- l'ensemble $\{x_5, x_6\}$ forme la 2^{ème} composante connexe, on la note C_2 .

On constate que les sommets de C_1 n'ont pas de relation de connexité avec les sommets de C_2 .

2-3 Le graphe connexe:

Un graphe $G=(X,U)$ est dit graphe connexe si tous ses sommets ont deux à deux la relation de connexité; autrement dit, si G contient une seule composante connexe.

Un graphe est connexe \Leftrightarrow il possède une seule composante connexe.

2-4 La recherche des composantes connexes:

Les composantes connexes d'un graphe se déterminent en utilisant un algorithme de marquage simple.

Le principe:

Soit $G=(X,U)$ un graphe orienté. L'idée de cet algorithme est la suivante: pour un sommet quelconque x du graphe G, il s'agit de trouver toutes les chaînes reliant ce sommet aux autres sommets du graphe; ainsi les sommets reliés au sommet x par une chaîne forment la composante connexe qui contient le sommet x .

Enoncé:

Données: un graphe $G=(X,U)$

Résultat: le nombre k de composantes connexes de G ainsi que la liste $\{C_1, C_2, \dots, C_k\}$ de ses composantes connexes.

(0) Initialisation: $k=0$, $W=X$.

(1)

(1.1) Choisir un sommet de W et le marquer d'un signe (+), puis marquer tous ses voisins d'un (+). On continue cette procédure jusqu'à ce qu'on ne puisse plus marquer de sommets.

(1.2) Poser $k=k+1$ et C_k l'ensemble des sommets marqués.

(1.3) Retirer de W les sommets de C_k et poser $W=W-C_k$.

(1.4) On teste si $W=\emptyset$.

- Si oui terminer aller à (2).

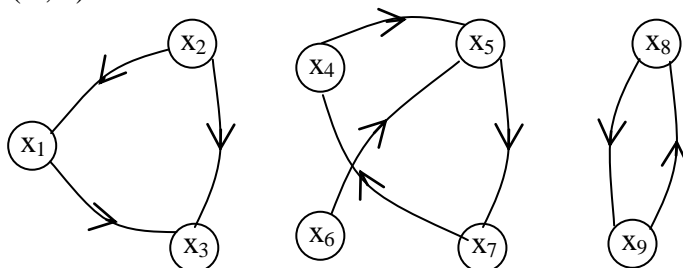
- Si non aller à (1)

(2) Le nombre de composantes connexes de (G) est k .

Chaque ensemble C_i , $i=1, \dots, k$ correspond aux sommets d'une composante connexe de (G) .

Application:

Soit le graphe $G=(X,U)$ suivant:

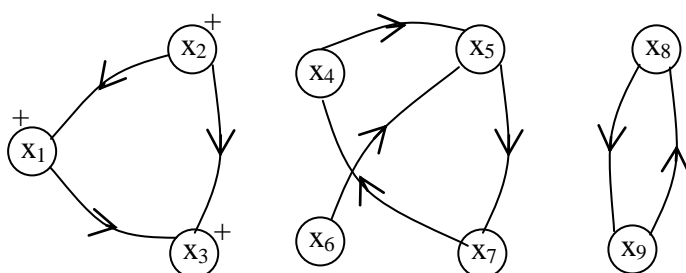


Le graphe (G) n'est pas connexe, car il n'existe pas de chaîne reliant les sommets x_3 et x_8 .

On applique l'algorithme de marquage précédent pour déterminer les composantes connexes:

Initialisation: $k=0$, $W = X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$

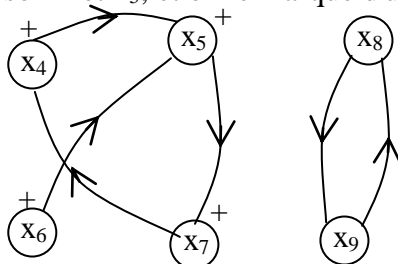
Itération1: on choisit dans W le sommet x_2 , et on le marque d'un signe (+), on marque ensuite ses voisins x_1 et x_3 .



Soit $C_1 = \{x_1, x_2, x_3\}$ l'ensemble des sommets marqués.

On retire de W les sommets C_1 , on obtient: $W = \{x_4, x_5, x_6, x_7, x_8, x_9\} \neq \emptyset$

Itération2: on choisit dans W le sommet x_5 , et on le marque d'un signe (+), on marque ensuite ses voisins x_4, x_6 et x_7 .



Soit $C_2 = \{x_4, x_5, x_6, x_7\}$ l'ensemble des sommets marqués.

On retire de W les sommets C_2 , on obtient: $W = \{x_8, x_9\} \neq \Phi$

Itération3: on choisit dans W le sommet x_8 , et on le marque d'un signe (+), on marque ensuite son voisin x_4, x_6 et x_7 .

Soit $C_3 = \{x_8, x_9\}$ l'ensemble des sommets marqués.

Retirons de W les sommets C_3 , on obtient: $W = \Phi$ terminer.



Le nombre de composantes connexes de (G) est 3, elles sont:

$C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6, x_7\}$ et $C_3 = \{x_8, x_9\}$

3- La forte connexité:

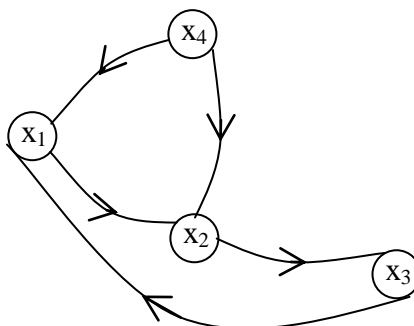
3-1 La notion de forte connexité:

On définit la forte connexité dans un graphe par une relation entre deux sommets de la manière suivante:

Deux sommets x et y ont une relation de forte connexité \Leftrightarrow il existe un chemin de x à y et un chemin de y à x ; ou bien $x=y$.

Exemple:

Soit le graphe $G=(X,U)$ suivant:



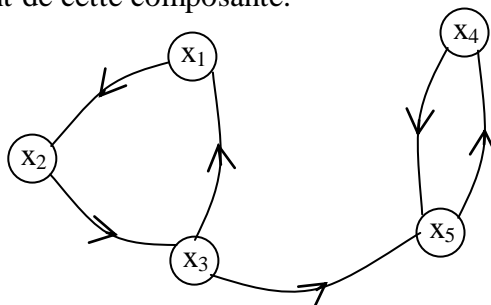
- On a un chemin reliant le sommet x_1 et x_3 et un chemin reliant le sommet x_3 au sommet x_1 alors x_1 et x_3 ont une relation de forte connexité.
- On a un chemin reliant le sommet x_4 à x_3 , mais on n'a pas de chemin reliant le sommet x_3 au sommet x_4 alors x_4 et x_3 n'ont pas de relation de forte connexité.

3-2 Les composantes fortement connexes:

On appelle composante fortement connexe un ensemble de sommets, qui ont deux à deux la relation de forte connexité, de plus tout sommet en dehors de la composante n'a pas de relation de forte connexité avec aucun élément de cette composante.

Exemple:

Soit le graphe $G=(X,U)$ suivant:



- Les sommets x_1, x_2, x_3 ont deux à deux la relation de forte connexité, donc l'ensemble $\{x_1, x_2, x_3\}$ forme ainsi la 1^{ère} composante fortement connexe, on la note C_1 .
- L'ensemble $\{x_4, x_5\}$ forme la composante fortement connexe, on la note C_2 .

On constate que les sommets de C_1 n'ont pas de relation de forte connexité avec les sommets de C_2 .

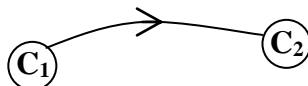
3-3 Le graphe réduit:

On appelle "graphe réduit" du graphe $G=(X,U)$, le graphe $G_r=(X_r,U_r)$ dont:

- Les sommets sont représentés par les composantes fortement connexes C_i du graphe G .
- Les arcs (x,y) dans le graphe G avec le sommet x appartenant à C_i et le sommet y appartenant à C_j , alors il existera un arc (C_i,C_j) dans le graphe réduit G_r .

Exemple:

Pour le graphe précédant, on a:



3-4 Le graphe fortement connexe:

Un graphe G est dit fortement connexe si tous ses sommets ont deux à deux la relation de forte connexité, autrement dit si G contient une seule composante fortement connexe.

Exemple:

- le graphe G précédant contient deux composantes fortement connexes $C_1 = \{x_1, x_2, x_3\}$ et $C_2 = \{x_4, x_5\}$, le graphe n'est pas fortement connexe.
- Si on ajoute l'arc (x_5, x_3) au graphe précédant, le graphe obtenu contient une seule composante fortement connexe $C = \{x_1, x_2, x_3, x_4, x_5\}$, alors il est fortement connexe.

3-5 La recherche des composantes fortement connexes:

Soit G un graphe orienté, un moyen de vérifier si ce graphe est fortement connexe est d'appliquer un algorithme de marquage suivant, qui permet de déterminer toutes ses composantes fortement connexes.

Le principe:

L'idée de cet algorithme est de parcourir le graphe à partir d'un sommet x dans le sens direct (en suivant les flèches des arcs) et dans le sens opposé (en suivant les flèches des arcs en sens inverse). Le premier ensemble obtenu regroupe les sommets accessibles à partir de x , le deuxième regroupe les sommets qui peuvent atteindre x . L'intersection de ces deux ensembles forme l'ensemble des sommets qui à la fois peuvent atteindre x et sont accessibles à partir de x . On obtient ainsi la composante fortement connexe qui contient le sommet x .

Enoncé:

Données: un graphe orienté $G=(X,U)$.

Résultat: le nombre k de composantes fortement connexes de G ainsi que la liste $\{C_1, C_2, \dots, C_k\}$ de ses composantes fortement connexes.

(0) Initialisation: $k=0$, $W=X$.

(1)

- (1.1) Choisir un sommet de W et le marquer d'un signe (+) et (-).
- (1.2) Marquer tous les successeurs directs et indirects de x avec (+).
- (1.3) Marquer tous les prédécesseurs directs et indirects de x avec (-).
- (1.4) Poser $k=k+1$ et C_k l'ensemble des sommets marqués avec (+) et (-).
- (1.5) Retirer de W les sommets de C_k et effacer toutes les marques; on pose $W=W-C_k$.
- (1.6) On teste si $W=\emptyset$.

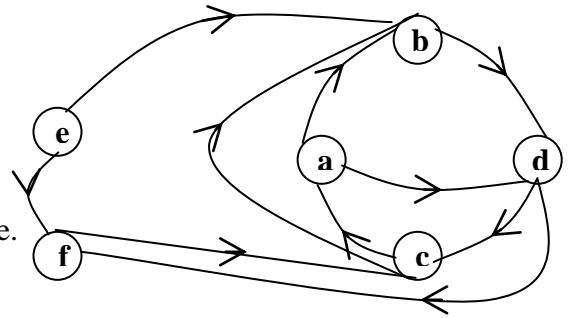
- Si oui terminer aller à (2).
- Si non aller à (1)

(2) Le nombre de composantes fortement connexes de (G) est k . Chaque ensemble C_i , $i=1, \dots, k$ correspond aux sommets d'une composante fortement connexe de (G) .

Application:

Soit le graphe $G=(X,U)$ suivant:

Du graphe (G), on constate qu'il y a un chemin reliant le sommet e à b, mais on n'a pas de chemin reliant le sommet b à e, alors e et b n'ont pas une relation de forte connexité, donc il existe plus d'une composante fortement connexe, on conclut que le graphe n'est pas fortement connexe.



On applique l'algorithme de marquage précédent pour déterminer les composantes fortement connexes:

Initialisation: $k=0$, $W = X = \{a, b, c, d, e, f\}$

Itération1: on choisit dans W, le sommet $\{a\}$ et on le marque par (+) et (-), on marque ensuite:

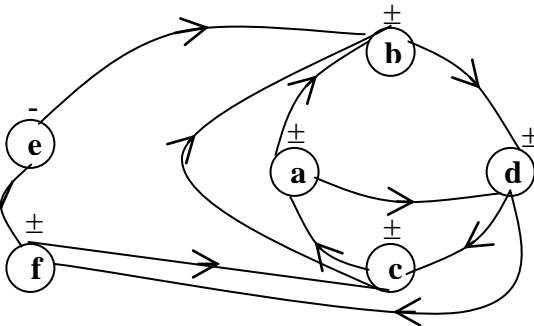
- Suivant l'orientation des arcs, les sommets b et d (successeurs directs du sommets a) avec le signe (+), ensuite les sommets c et f (successeurs indirect du sommet a) avec le signe (+).
- Suivant l'orientation des arcs en sens inverse, le sommet c (prédécesseur direct du sommet a) avec le signe (-), ensuite les sommets b,d,e,f (prédécesseurs indirects du sommet a) avec le signe (-).

Soit $C_1 = \{a, b, c, d, f\}$ l'ensemble des sommets marqués à la fois par (+) et (-).

On retire de W, les sommets de C_1 , on obtient:

$W = \{e\} \neq \Phi$

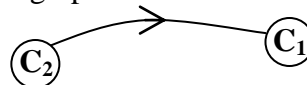
Itération2: on marque le sommet e d'un signe (+) et (-), on constate qu'il n'y a pas d'autres sommets à marquer. Alors $C_2 = \{e\}$ l'ensemble des sommets marqués à la fois par (+) et (-).



On retire de W les sommets de C_2 on obtient: $W = \Phi$, terminer.

Le nombre de composantes fortement connexes de (G) est 2, elles sont: $C_1 = \{a, b, c, d, f\}$ et $C_2 = \{e\}$.

Au graphe (G), on fait correspondre un graphe réduit noté G_r représenté comme suit:



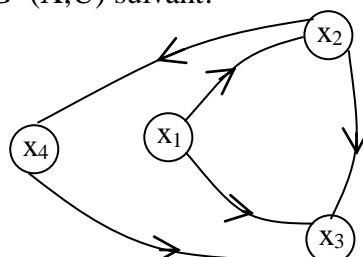
4- La mise en ordre d'un graphe connexe ou la recherche d'un circuit:

4-1 La mise en ordre d'un graphe connexe (l'ordonnement d'un graphe):

Le principe:

Ordonner un graphe revient à disposer dans un certain ordre ses sommets tels que les arcs soient dans le même sens. On définit ainsi les différents niveaux des sommets du graphe.

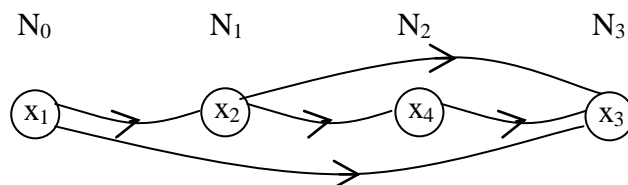
Exemple: Soit le graphe $G=(X,U)$ suivant:



Les niveaux des sommets du graphe sont définis comme suit:

- Le niveau nul, noté N_0 détermine les sommets du graphe n'ayant pas de prédécesseurs ($\Gamma_G^-(x) = \Phi$). Dans le graphe G le sommet x_1 n'a pas de prédécesseurs d'où $N_0 = \{x_1\}$.
- Le premier niveau noté N_1 définit les sommets du graphe dont tous les prédécesseurs appartiennent à N_0 . Dans le graphe G le sommet x_2 admet comme prédécesseurs le sommet x_1 qui appartient au niveau N_0 d'où $N_1 = \{x_2\}$.
- Le deuxième niveau noté N_2 définit les sommets du graphe dont tous les prédécesseurs appartiennent à $N_0 \cup N_1$. Dans le graphe G le sommet x_4 admet comme prédécesseurs le sommet x_2 qui appartient à $N_0 \cup N_1$ d'où $N_2 = \{x_4\}$.
- Le troisième niveau noté N_3 définit les sommets du graphe dont tous les prédécesseurs appartiennent à $N_0 \cup N_1 \cup N_2$. Le sommet x_3 admet comme prédécesseurs le sommet x_1, x_2 et x_4 qui appartient à $N_0 \cup N_1 \cup N_2$ d'où $N_3 = \{x_3\}$.

Le graphe ordonné de G est:



Le procédé:

La mise en ordre d'un graphe connexe $G=(X,U)$ ou l'ordonnancement d'un graphe se traduit par l'algorithme suivant:

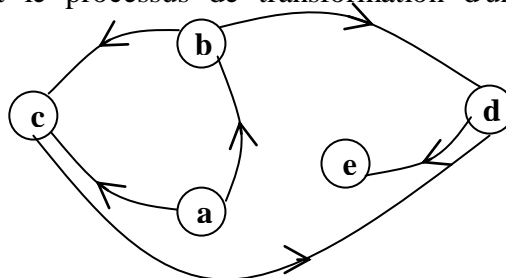
Données: un graphe $G=(X,U)$.

Résultat: les différents niveaux de sommets du graphe ainsi que le graphe ordonné de G .

- (0) On détermine le dictionnaire des prédécesseurs du graphe G formé par le couple $(W, \Gamma_G^-(x))$.
- (1) On repère dans le dictionnaire des prédécesseurs du graphe les sommets n'ayant pas de prédécesseurs ($\Gamma_G^-(x) = \Phi$).
 - (1.1) On pose N_0 l'ensemble des sommets du graphe n'ayant pas de prédécesseurs, on l'appelle niveau nul.
 - (1.2) On barre dans la colonne de $\Gamma_G^-(x)$ tous les sommets de niveaux nul N_0 , on obtient une nouvelle colonne $\Gamma_{G_1}^-(x)$, avec G_1 le sous-graphe engendré par X/N_0 .
- (2) On repère dans la nouvelle colonne $\Gamma_{G_1}^-(x)$ les sommets n'ayant pas de prédécesseurs ($\Gamma_{G_1}^-(x) = \Phi$).
 - (2.1) On pose N_1 l'ensemble des sommets du graphe n'ayant pas de prédécesseurs.
 - (2.2) On barre dans la colonne de $\Gamma_{G_1}^-(x)$ tous les sommets de niveaux N_1 , on obtient une nouvelle colonne $\Gamma_{G_2}^-(x)$, avec G_2 le sous-graphe engendré par $X / N_0 \cup N_1$.

On continue le même procédé jusqu'à ce qu'on termine le graphe et on représente ainsi le graphe ordonné par niveaux de G .

Exemple: Soit le graphe $G=(X,U)$ représentant le processus de transformation d'une matière première a dans un atelier.



1) Soit le dictionnaire des prédécesseurs du G:

Le sommet a n'a pas de prédécesseurs,
il est donc de niveau nul $N_0 = \{a\}$

X	$\Gamma_G^-(x)$
a	Φ
b	a
c	a-b
d	c-b
e	d

on barre le sommet a de la colonne $\Gamma_G^-(x)$, on obtient la nouvelle colonne $\Gamma_{G_1}^-(x)$.

2) Le sommet b n'a plus de prédécesseurs, il est donc de niveau un $N_1 = \{b\}$

X	$\Gamma_{G_1}^-(x)$
a	-
b	Φ
c	b
d	c- b
e	d

On barre le sommet b de la colonne $\Gamma_{G_1}^-(x)$, on obtient la nouvelle colonne $\Gamma_{G_2}^-(x)$.

3) le sommet c n'a plus de prédécesseurs, il est donc de niveau deux $N_2 = \{c\}$

X	$\Gamma_{G_2}^-(x)$
a	-
b	-
c	Φ
d	c
e	d

On barre le sommet c de la colonne $\Gamma_{G_2}^-(x)$, on obtient le nouveau dictionnaire.

4) le sommet d n'a plus de prédécesseurs, il est donc de niveau trois $N_3 = \{d\}$

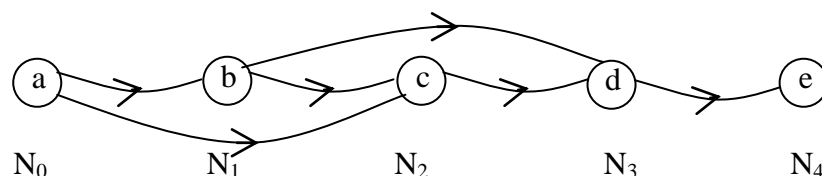
X	$\Gamma_{G_3}^-(x)$
a	-
b	-
c	-
d	Φ
e	d

On barre le sommet d de la colonne $\Gamma_{G_3}^-(x)$, on obtient le nouveau dictionnaire.

4) le sommet e n'a plus de prédécesseurs, il est donc de niveau quatre $N_4 = \{e\}$

X	$\Gamma_{G_4}^-(x)$
a	-
b	-
c	-
d	-
e	Φ

On a examiné tous les sommets du graphe G, on a ainsi le graphe ordonné de G représenté ci dessous.



4.2 La recherche d'un circuit dans un graphe connexe:

A une étape donnée de l'ordonnancement d'un graphe, la définition des niveaux se bloque (i.e il n'existe pas de sommets n'ayant pas de prédécesseurs), donc la mise en ordre du graphe est impossible, on dit alors que le graphe possède un circuit.

Comment faire pour déterminer ce circuit?

On applique la procédure suivante:

Soit $(X, \Gamma_{Gi}^-(x))$ le dernier dictionnaire des prédécesseurs du graphe G où l'application de l'algorithme de mise en ordre du graphe G est bloquée.

1) On repère dans la colonne $\Gamma_{Gi}^-(x)$ un sommet quelconque x_0 , ce sommet nous renvoi à sa position dans la colonne X.

2) On lit à nouveau dans la colonne $\Gamma_{Gi}^-(x)$ un nouveau sommet x_1 correspondant au prédécesseur de x_0 , lequel nous renvoi à sa position dans la colonne de X et ainsi de suite.

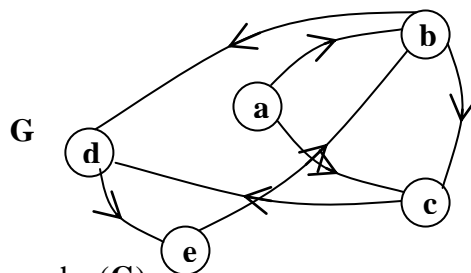
Cette procédure s'arrête lorsque dans la suite des sommets lus, un sommet se répète. Le circuit est alors entre ces deux mêmes sommets de la suite.

Remarque:

L'écriture de la suite sommets se fait de gauche à droite, c'est-à-dire $(x_0, x_1, \dots, x_i, \dots, x_p, \dots, x_i)$ et, la lecture du circuit se fait de droite à gauche.

Application:

Soit le graphe $G=(X,U)$ suivant:



1) Soit le dictionnaire des prédécesseurs du graphe (G):

Le sommet a n'a pas de prédécesseurs, il est donc de niveau nul $N_0 = \{a\}$

X	$\Gamma_G^-(x)$
a	Φ
b	a -e
c	a -b
d	c-b
e	d

on barre le sommet a de la colonne $\Gamma_G^-(x)$, on obtient le dictionnaire suivant:

X	$\Gamma_{G1}^-(x)$
a	-
b	e
c	b
d	c-b
e	d

2) A cette étape, on ne peut pas déterminer le niveau N_1 car il n'y a pas de sommets n'ayant pas de prédécesseurs, donc le graphe n'est pas ordonnable; G contient alors un circuit.

Pour déterminer ce circuit:

- On choisit un sommet dans $\Gamma_{G_1}^-(x)$, soit e ce sommet, celui-ci nous renvoie à sa ligne dans X .

X	$\Gamma_{G_1}^-(x)$
a	-
b	e
c	b
d	c-b
e	d

- On choisit un sommet dans la ligne de e , soit d ce sommet, celui-ci nous renvoie à sa ligne dans X .

X	$\Gamma_{G_1}^-(x)$
a	-
b	e
c	b
d	b-c
e	d

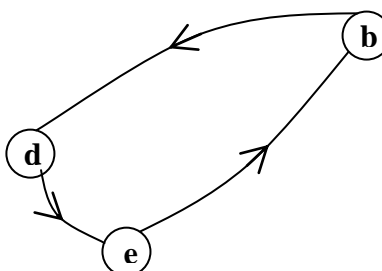
- On choisit un sommet dans la ligne de d , soit b ce sommet, celui-ci nous renvoie vers la ligne dans X .

X	$\Gamma_{G_1}^-(x)$
a	-
b	e
c	b
d	b-c
e	d

- On choisit un sommet dans la ligne de b , soit e ce sommet. On arrête la procédure car le sommet s'est répété.

X	$\Gamma_{G_1}^-(x)$
a	-
b	e
c	b
d	e-b
e	d

La suite des sommets trouvée est la suivante: $e d b e$ (la lecture du circuit se fait de droite à gauche), on obtient ainsi, le circuit: $(ebde)$



CHAPITRE 3: ARBRES ET ARBORESCENCES

1- Arbres et arborescences:

☒ Propriété 1:

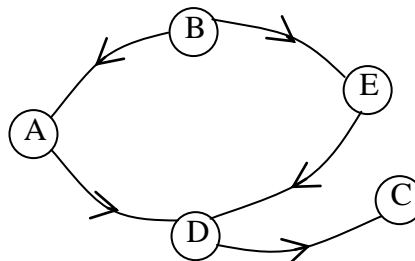
Soit n le nombre de sommets d'un graphe $G=(X,U)$, et m le nombre de ses arcs.

- Si G est connexe $\Rightarrow m \geq n-1$
- Si G est sans cycles $\Rightarrow m \leq n-1$

Exemple:

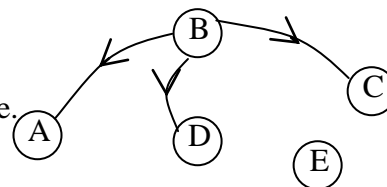
- Soit le graphe $G=(X,U)$ suivant:

Le graphe G est connexe et contient un cycle $C=(A,B,E,D)$.



- Soit le graphe $G=(X,U)$ suivant:

Dans le graphe G , le nombre d'arcs ($m=3$) est inférieur au nombre de sommets ($n-1=4$), alors le graphe G n'est pas connexe.



☒ Définition d'un arbre:

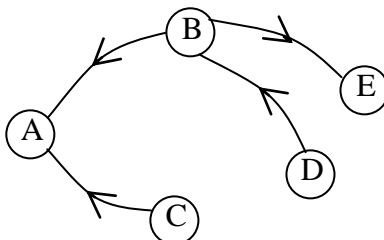
Un arbre, est par définition, un graphe connexe et sans cycle.

Remarque:

D'après la définition, un arbre est un graphe simple sans boucle, ayant $(n-1)$ arcs.

Exemple:

Le graphe $G=(X,U)$ est un arbre.



☒ Propriété 2:

Soit $G=(X,U)$ un graphe sur $n=|X| \geq 2$ sommets. Les propriétés suivantes sont équivalentes et caractérisent un arbre:

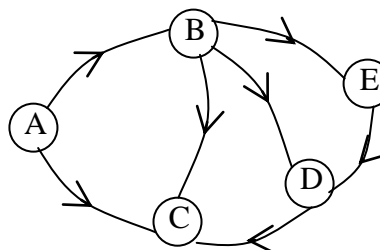
- (i) G est connexe et sans cycle.
- (ii) G est connexe et est minimal pour cette condition (si on supprime un arc de G , il ne sera plus connexe).
- (iii) G est connexe et possède $(n-1)$ arcs.
- (iv) G est sans cycle et, est maximal pour cette propriété (si on ajoute un arc à G , il possédera un cycle).
- (v) G est sans cycle et possède $(n-1)$ arcs.
- (vi) Entre chaque couple de sommets, il existe une et une seule chaîne les reliant.

Remarque:

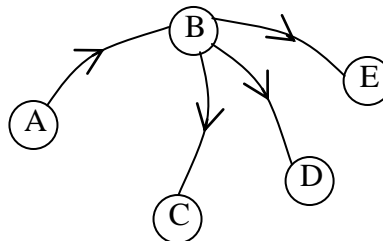
D'un graphe connexe $G=(X,U)$ on peut extraire un graphe partiel qui est un arbre.

Exemple:

Soit le graphe $G=(X,U)$ suivant:



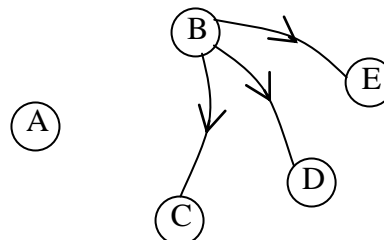
Soit $W = \{AB, BC, BE, BD\}; W \subset U$. On définit ainsi le graphe partiel engendré par W , $G_W = (X, W)$ qui est un arbre.



Dans le graphe partiel G_W :

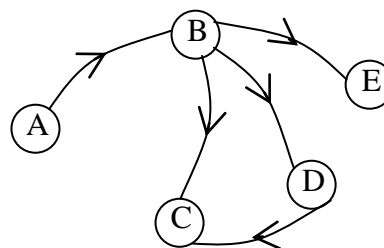
- En vertu de la propriété 1, les conditions (i), (iii) et (v) sont vérifiées.
- Si on supprime un arc de G_W , par exemple l'arc (AB), le graphe obtenu n'est plus connexe. La condition (ii) est vérifiée.

Le graphe obtenu après suppression de l'arc (AB) n'est pas connexe.



- Si on ajoute un arc à G_W , par exemple l'arc (DC), le graphe obtenu possèdera un cycle. La condition (iii) est vérifiée.

Le graphe obtenu après l'ajout de l'arc (DC), possède un cycle.



- Pour tout couple de sommets, il existe une chaîne et une seule les reliant. La condition (vi) est donc, vérifiée.

☒ Définition d'une forêt:

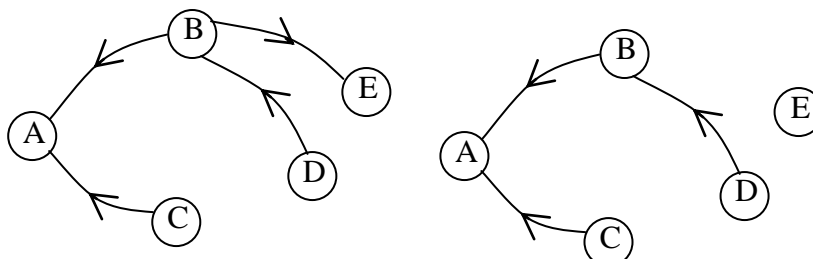
Une forêt est un graphe dont chaque composante connexe est un arbre. C'est-à-dire un graphe sans cycle.

Remarque:

- La forêt est obtenue si on relâxe (relâche) la contrainte de connexité dans un arbre, c'est-à-dire, si on supprime un arc dans un arbre.
- Tout graphe partiel d'un arbre est une forêt.

Exemple:

Dans le graphe de la figure suivante, si on relâche la contrainte de connexité en supprimant par exemple l'arc BE, on obtient une forêt.



La forêt obtenue est un graphe partiel engendré par l'ensemble des arcs $W = \{BA, CA, DB\}$.

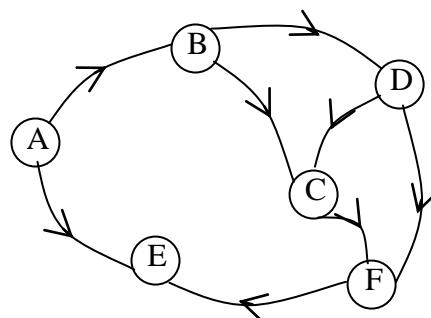
☒ Définition d'une racine:

Un sommet s d'un graphe G est une "racine" (resp, une "antiracine") s'il existe un chemin joignant s à chaque sommet du graphe G (resp. joignant chaque sommet de G à s) à l'exception du sommet lui-même. C'est-à-dire : $\forall x \in X - \{s\}$, il existe un chemin de s à x (resp de x à s).

Exemple:

Soit le graphe $G=(X,U)$ suivant:

- Le sommet A est une racine du graphe G.
- Le sommet E est une antiracine du graphe G.



☒ Définition d'une arborescence:

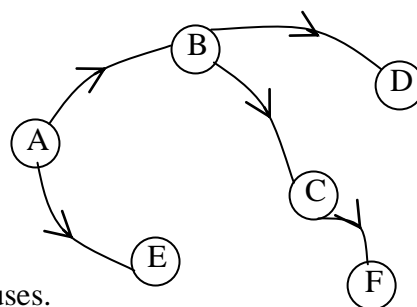
Un graphe $G=(X,U)$, avec $|X| = n \geq 2$ sommets est une arborescence de racine s si:

- G est un arbre.
- s est une racine.

Exemple:

Soit $W=(X,U)$ l'arbre suivant:

W est un arbre admettant le sommet A comme racine alors W est une arborescence.



La notion d'arborescence a des applications très nombreuses.

- Découpage d'un livre (chapitre, paragraphe...)
- Arbre généalogique (sommets membre de famille et les arcs les liens de parenté directs)
- ...

Remarque:

Une arborescence est un arbre mais la réciproque est fausse.

☒ Définition anti-arborescence:

Un graphe $G=(X,U)$ sur $n = |X| \geq 2$ sommets est une anti-arborescence admettant le sommet s comme antiracine si:

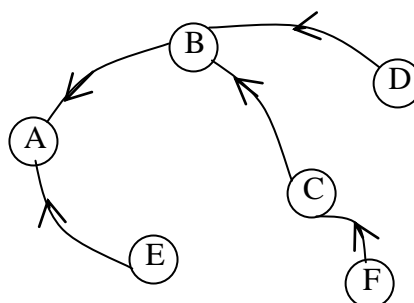
- G est un arbre.
- s est une antiracine de G .

Remarque:

Si on inverse le sens des arcs d'une arborescence, on obtient une anti-arborescence.

Exemple:

Si on inverse le sens des arcs de l'arborescence précédente, on aura l'anti-arborescence suivante:



Le sommet A est une antiracine.

2- Le problème de recherche d'un arbre de poids minimum (minimum tree of pods)

Si on associe à chaque arc d'un graphe $G=(X,U)$ une valeur (un poids). Le problème de l'arbre de coût minimum consiste à trouver un sous graphe qui est un arbre, dont la somme des poids des arcs est minimale.

Par exemple: minimiser le coût d'installation des lignes téléphoniques dans une localité peut être représenté comme un problème de recherche d'un arbre de coût minimum. En effet, on veut relier tous les points de la localité sans avoir de lignes inutiles, d'où la recherche d'un arbre. Ensuite on veut avoir un coût d'installation minimum, alors on associera à chaque possibilité d'installation d'une ligne le coût nécessaire et on cherchera à minimiser le coût total de toute l'installation.

Plusieurs algorithmes existent pour résoudre les problèmes de recherche d'un arbre de poids minimum, nous ne présenterons que les méthodes qui paraissent les plus célèbres.

2-1 Algorithme de Sollin:

Le principe:

L'idée de l'algorithme de Sollin est de démarrer par un graphe partiel sans arêtes A (comportant n composantes connexes), puis on va choisir une composante connexe C , on va choisir pour C l'arrête adjacente de poids minimum, on ajoute cette arrête au A . Ce processus est répété tant que le nombre de composantes connexes est différent de un.

Enoncé:

Données: un graphe valué $G=(X,U,c)$.

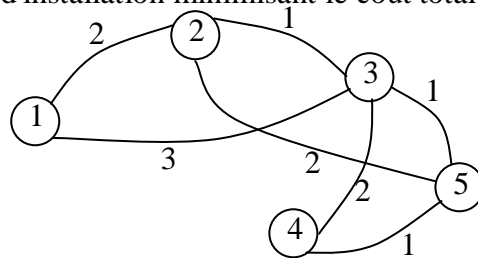
Résultat: un arbre A

- (0) Initialisation: soit A le graphe partiel de G sans arêtes;
- (1) Tant Que (A compte plus d'une composante connexe) faire
 - (1-1) Choisir une composante connexe C_i de A ;
 - (1-2) Ajouter à A une arête de poids minimal adjacente de C_i reliant C_i à une autre composante connexe de A ;
- (2) Fin Tant Que

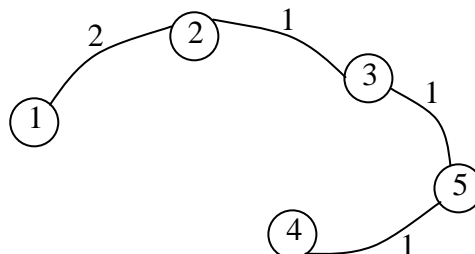
Application:

Soit $G=(X,U)$ un graphe connexe représentant le projet d'installation de lignes téléphoniques. Les poids représentent le coût d'installation des lignes.

On veut donner un plan d'installation minimisant le coût total de l'installation.



Composantes connexes	Composante choisie	Arête choisie
$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$	$\{3\}$	1-5
$\{1\}, \{2\}, \{3,5\}, \{4\}$	$\{1\}$	1-2
$\{1,2\}, \{3,5\}, \{4\}$	$\{3,5\}$	3-2
$\{1,2,3,5\}, \{4\}$	$\{1,2,3,5\}$	5-4
$\{1,2,3,5,4\}$	Une seule composante connexe, on s'arrête	



2-2 Algorithme de kruskal:

Le principe:

L'idée de l'algorithme de Kruskal est tout d'abord de numéroter les arcs par ordre des poids croissants. Ensuite de construire progressivement l'arbre A en rajoutant dans leur ordre, les arcs un par un. Un arc est ajouté seulement si son adjonction à A ne détermine pas de cycle, c'est-à-dire si A ne perd pas sa notion d'arbre, sinon on passe à l'arc suivant dans l'ordre de la numérotation.

Enoncé:

Données: un graphe valué $G=(X,U,c)$.

Résultat: ensemble d'arcs W

(3) Initialisation: numéroter les arcs de G dans l'ordre des poids croissants:

$c(u_1) \leq c(u_2) \leq \dots \leq c(u_m)$. Soit $W = \emptyset$; $i=1$.

(4) Si $(X, W \cup \{u_i\})$ contient un cycle aller en (3)

Sinon aller en (2)

(5) On pose $W := W \cup \{u_i\}$ aller (3)

(6) Si $i=m$ terminé, $A=(X,W)$ est l'arbre de poids minimum $c(W) = \sum c(u_i)$ pour $u_i \in W$

Sinon $i:=i+1$ aller en (1)

L'algorithme s'arrête lorsque le nombre d'arcs retenus est égal à $n-1$.

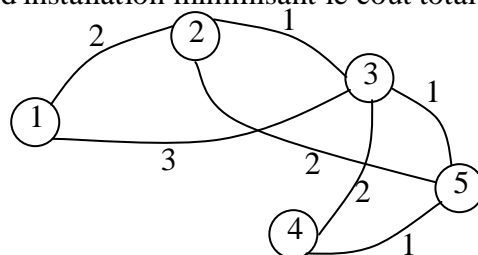
Remarque:

La solution du problème de recherche de l'arbre de poids maximum s'obtient en utilisant le même algorithme de l'arbre de poids minimum après avoir multiplié les poids des arcs par (-1) .

Application:

Soit $G=(X,U)$ un graphe connexe représentant le projet d'installation de lignes téléphoniques. Les poids représentent le coût d'installation des lignes.

On veut donner un plan d'installation minimisant le coût total de l'installation.



Initialisation: On ordonne les arêtes du graphe selon les poids croissants:

i	1	2	3	4	5	6	7
u_i	(2,3)	(3,5)	(4,5)	(3,4)	(2,5)	(1,2)	(1,3)
$c(u_i)$	1	1	1	2	2	2	3

Soit $W = \emptyset$; $i=1; m=7$

- Itération 1:

On a : $u_1=(2,3)$; Soit $W = W \cup \{u_1\} = \{(2,3)\}$

Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors

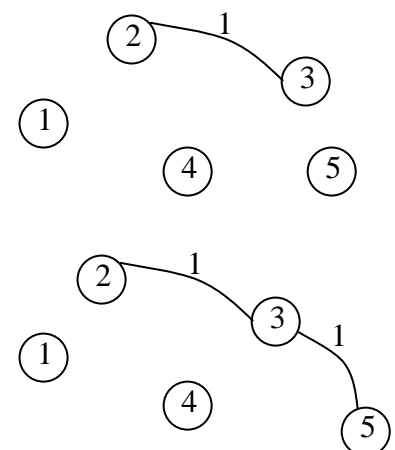
$W = \{(2,3)\}$ $|W| = 1$. On a: $i \neq m$ alors on pose $i=i+1=2$.

- Itération 2:

On a : $u_2=(3,5)$; Soit $W = W \cup \{u_2\} = \{(2,3), (3,5)\}$

Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors

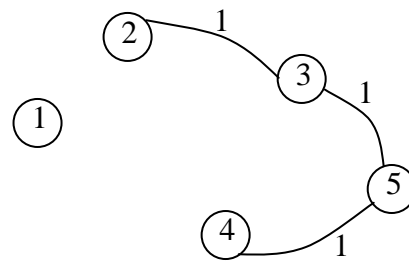
$W = \{(2,3), (3,5)\}$ $|W| = 2$. On a: $i \neq m$ alors on pose $i=i+1=3$.



- **Itération 3:**

On a : $u_3=(4,5)$; Soit $W = W \cup \{u_3\} = \{(2,3), (3,5), (4,5)\}$

Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors $W = \{(2,3), (3,5), (4,5)\} \quad |W| = 3$. On a: $i \neq m$ alors on pose $i=i+1=4$.

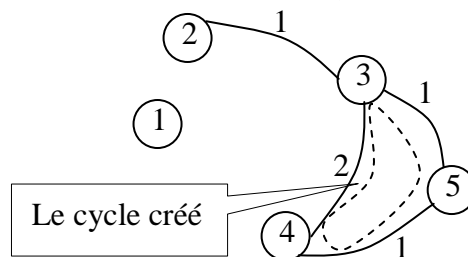


- **Itération 4:**

On a : $u_4=(3,4)$; Soit $W = W \cup \{u_4\} = \{(2,3), (3,5), (4,5), (3,4)\}$

Le graphe (X,W) ci contre contient un cycle,

On a: $i \neq m$ alors on pose $i=i+1=5$.

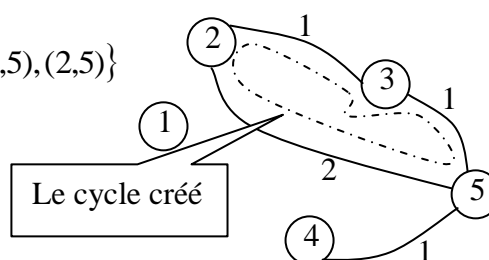


- **Itération 5:**

On a : $u_5=(2,5)$; Soit $W = W \cup \{u_5\} = \{(2,3), (3,5), (4,5), (2,5)\}$

Le graphe (X,W) ci contre contient un cycle,

On a: $i \neq m$ alors on pose $i=i+1=6$.



- **Itération 6:**

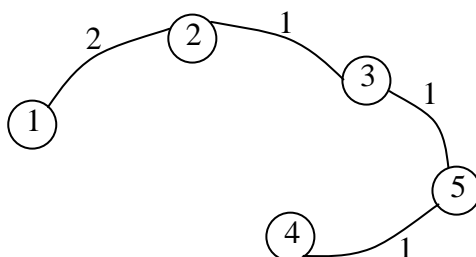
On a : $u_6=(1,2)$; Soit $W = W \cup \{u_6\} = \{(2,3), (3,5), (4,5), (1,2)\}$

Le graphe (X,W) ci contre ne contient pas de cycle, on pose alors

$W = \{(2,3), (3,5), (4,5), (1,2)\} \quad |W| = n - 1 = 5 - 1 = 4$, terminé

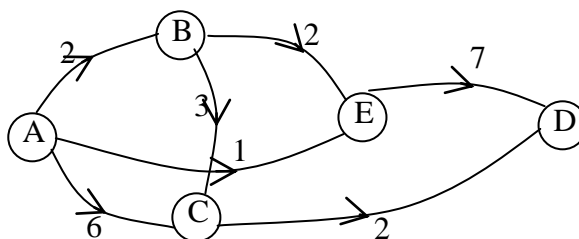
L'arbre du poids minimum est représenté par le graphe $A=(X,W)$.

Le coût total de toute l'installation est: $c(W)=c(u_1)+c(u_2)+ c(u_3)+ c(u_6)=1+1+1+2=5$.



3- Le problème de recherche d'un plus court chemin:

La figure suivante représente les routes à sens unique reliant deux points A et D, les évaluations des arcs représentent des distances kilométriques.



Il existe plusieurs chemins reliant le sommet A au sommet D, mais on veut trouver celui de distance minimale.

On considère le tableau suivant qui donne tous les chemins issus du sommet A et leurs longueurs:

Chemin issu du sommet A	La longueur du chemin km
AB	2
ABC	5
AC	6
ABE	4
AE	1
ABCD	7
ABED	11
ACD	8
AED	8

Il en résulte que le plus court chemin reliant A à D est (ABCD), il a pour longueur 7km.

Le principe de la recherche d'un plus court chemin issu d'un sommet x vers un autre sommet y dans un graphe G est de trouver tous les chemins élémentaires reliant x à y et d'en choisir le plus court.

3-1 Définitions:

a) Réseau:

Un réseau est un graphe $G=(X,U)$ muni d'une application $d : U \rightarrow \mathbb{R}$ qui à chaque arc fait correspondre sa longueur $d(u)$, on note un tel réseau par: $R=(X,U,d)$. En pratique, $d(u)$ peut matérialiser un coût, une distance, une durée, ..etc.

b) La longueur d'un chemin dans un réseau:

La longueur d'un chemin C dans un réseau R est égale à la somme des longueurs des arcs comptés dans leur ordre de multiplicité dans le chemin (le nombre de fois qu'on parcourt ces arcs), on le note par $l(C) = \sum_{u \in C} d(u)\eta(u)$.

On définit de la même manière, la longueur d'un cycle, d'un circuit et d'une chaîne.

Exemple:

Dans le réseau précédent, soit le chemin $C=(A,B,C,D)$. Sa longueur est::

$$l(C) = \sum_{u \in C} d(u)\eta(u) = d(AB)\eta(AB) + d(BC)\eta(BC) + d(CD)\eta(CD) = 2*1 + 3*1 + 2*1 = 7$$

Dans le chemin C, l'ordre de multiplicité de chaque arc est de 1.

c) Plus courte distance:

Si un plus court chemin (chemin de longueur minimale) d'un sommet s à un sommet x existe dans un réseau. La longueur de ce plus court chemin sera appelée " plus courte distance de s à x " et se notera $\pi(x)$.

Exemple:

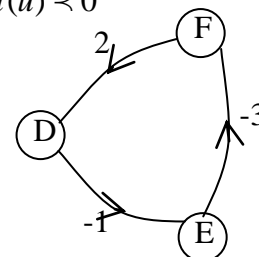
Dans le réseau précédent, il existe deux chemins de A à B. Le plus court chemin est AB de longueur 1, donc la plus courte distance de A à B est $\pi(B) = 1$.

d) Circuit absorbant:

Un circuit est dit absorbant si sa longueur est négative: $l(C) = \sum_{u \in C} d(u) < 0$

Exemple:

Dans le réseau suivant, le circuit $C=(D,E,F,D)$ est absorbant car $l(C) = -2 < 0$ et plus on contourne le circuit, plus sa longueur devient minimale.(tend vers $-\infty$) (dans ce cas on ne pourra pas déterminer un plus court chemin.



Dans cette leçon, on s'intéresse à la recherche d'un plus court chemin, joignant un sommet s donné à chaque sommet x du réseau $R=(X,U,d)$.

Ce type de problème admet une solution si et seulement si:

- s est une racine du réseau.
- Le réseau n'admet pas de circuit absorbant.

La solution de ce type de problème si elle existe est une arborescence des plus courts chemins admettant s comme racine.

3-2 Algorithme de recherche d'un plus court chemin:

Plusieurs algorithmes existent pour résoudre les problèmes de recherche des plus courts chemins dans un réseau, nous ne présenterons que les méthodes qui paraissent le plus performantes.

a) Algorithme de Bellman:

On applique cet algorithme pour la recherche d'une arborescence de plus courts chemins dans un réseau $R=(X,U,d)$ sans circuit.

Le principe:

L'idée de l'algorithme de Bellman, est de calculer de proche en proche, l'arborescence des plus courtes distances, issue du sommet s à un sommet donné p . On ne calcule la plus courte distance du sommet s à y , que si on a déjà calculé les plus courtes distances du sommet s à tous les prédécesseurs du sommet y .

Enoncé:

Données: un réseau $R=(X,U,d)$ sans circuit avec $d(u) \in R$.

Résultat: Arborescence de plus courtes distances A .

(0) Initialisation: Soit s un sommet de X , on pose $S = \{s\}$ et $\pi(s) = 0$ et $A = \Phi$.

(1) Chercher un sommet hors de S dont tous les prédécesseurs sont dans S .

- Si un tel sommet n'existe pas; Terminer

Dans ce cas soit $S=X$, ou le sommet s n'est pas une racine dans R .

- Si un tel sommet existe; Aller en (2)

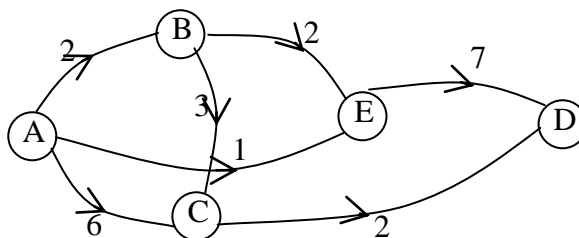
(2) On pose: $\pi(x) = \min\{\pi(I(u)) + d(u)\}$.

Soit u l'arc pour lequel $\pi(x) = \pi(I(u)) + d(u)$

$A := A \cup \{u\}; S := S \cup \{x\}$ Aller à (1)

Application:

Soit le réseau $R=(X,U,d)$ suivant:



Initialisation:

Soit A un sommet de X. on pose $S = \{A\}; \pi(s) = 0; A = \Phi$.

- **Itération 1:**

Le sommet B est hors de S et tous ses prédécesseurs sont dans S, alors:

$$\pi(B) = \pi(A) + d(A, B) = 0 + 2 \Rightarrow \dot{u} = (A, B)$$

$$S = S \cup \{B\} = \{A, B\}; A = A \cup \{(A, B)\} = \{(A, B)\}$$

- **Itération 2:**

Les sommets E et C sont hors de S et tous leurs prédécesseurs sont dans S, alors:

$$\pi(E) = \min(\pi(A) + d(A, E); \pi(B) + d(B, E)) = \min(0 + 1; 2 + 2) = 1 \Rightarrow \dot{u}_1 = (A, E)$$

$$\pi(C) = \min(\pi(A) + d(A, C); \pi(B) + d(B, C)) = \min(0 + 6; 2 + 3) = 5 \Rightarrow \dot{u}_2 = (B, C)$$

$$S = S \cup \{E, C\} = \{A, B, E, C\}; A = A \cup \{(A, E), (B, C)\} = \{(A, B), (A, E), (B, C)\}$$

- **Itération 3:**

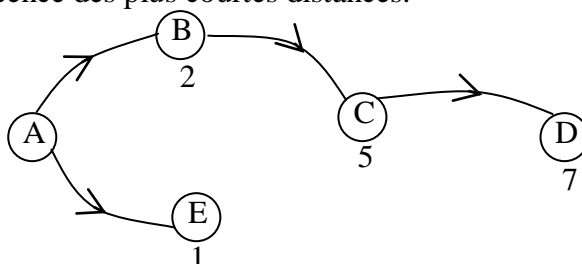
Les sommets D est hors de S et tous ses prédécesseurs sont dans S, alors:

$$\pi(D) = \min(\pi(E) + d(E, D); \pi(C) + d(C, D)) = \min(1 + 7; 5 + 2) = 7 \Rightarrow \dot{u} = (C, D)$$

$$S = S \cup \{D\} = \{A, B, E, C, D\}; A = A \cup \{(C, D)\} = \{(A, B), (A, E), (B, C), (C, D)\}$$

On a: $S=X$; Terminer.

On obtient donc, l'arborescence des plus courtes distances:



b) Algorithme de Dijkstra:

On applique cette algorithme pour déterminer une arborescence des plus courtes distances sur un réseau $R=(X,U,d)$, où les longueurs des arcs sont positives ou nulles ($d(u) \geq 0 \forall u \in U$).

Le principe:

L'idée de l'algorithme de Dijkstra est de calculer de proche en proche, l'arborescence des plus courtes distances, issue du sommet s à un sommet donné p. Une particularité de cet algorithme est que les distances s'introduisent dans l'ordre croissant.

Énoncé:

Données: Un réseau $R=(X,U,d)$ avec $d(u) \geq 0 \forall u \in U$

Résultat: Arborescence de plus courtes distances A.

(0) Initialisation: Soit s un sommet de X.

On pose : $S = \{s\}, \pi(s) = 0, \pi(x) = \infty, \forall x \in X / \{s\}, A(s) = \Phi$ et $\alpha = s$

/* α est le dernier sommet introduit dans S

(1) Examiner tous arcs u dont l'extrémité initiale est égale à $\alpha(I(u) = \alpha)$ et l'extrémité terminale n'appartient pas à S ($T(u) = y$ avec $y \notin S$).

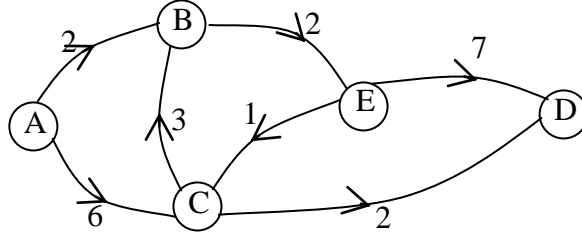
Si $\pi(\alpha) + d(u) < \pi(y)$, on pose $\pi(y) = \pi(\alpha) + d(u)$ et $A(y) = u$ Aller en (2).

(2) Choisir un sommet $z \notin S$ tel que $\pi(z) = \min\{\pi(y) / y \notin S\}$

- ✓ Si $\pi(z) = \infty$; Terminer. Le sommet s n'est pas une racine dans R.
- ✓ Si $\pi(z) < \infty$; on pose $\alpha = z$ et $S = S \cup \{\alpha\}$.
- ✓ Si $S=X$; Terminer. A définit l'arborescence des plus courts chemins issus de s.
- ✓ Si $S \neq X$ Aller en (1)

Application:

Soit le réseau $R=(X,U,d)$ suivant:



Initialisation:

Soit A un sommet de X. on pose $S = \{A\}$; $\pi(A) = 0$; $A(A) = \Phi$, $\alpha = A$.

x	A	B	C	D	E
$\pi(x)$	0	∞	∞	∞	∞

- **Itération 1:**

On examine les arcs (A,B) et (A,C)

$$\pi(A) + d((A,B)) = 0 + 2 = 2 < \pi(B) = \infty \Rightarrow \pi(B) = 2; A(B) = (A,B)$$

$$\pi(A) + d((A,C)) = 0 + 6 = 6 < \pi(C) = \infty \Rightarrow \pi(C) = 6; A(C) = (A,C)$$

x	A	B	C	D	E
$\pi(x)$	0	2	6	∞	∞

$$\pi(z) = \min\{\pi(y) / y \notin S\} = \min\{\pi(B), \pi(C), \pi(D), \pi(E)\} = \min\{2, 6, \infty, \infty\} = 2 = \pi(B) \text{ Donc } z=B$$

On pose alors: $\alpha = B$; $S = \{A, B\} \neq X$; $A = \{(A, B)\}$

- **Itération 2:**

On examine l'arc (B,E)

$$\pi(B) + d((B,E)) = 2 + 2 = 4 < \pi(E) = \infty \Rightarrow \pi(E) = 4; A(E) = (B, E)$$

x	A	B	C	D	E
$\pi(x)$	0	2	6	∞	4

$$\pi(z) = \min\{\pi(y) / y \notin S\} = \min\{\pi(C), \pi(D), \pi(E)\} = \min\{6, \infty, 4\} = 4 = \pi(E) \text{ Donc } z=E$$

On pose alors: $\alpha = E$; $S = \{A, B, E\} \neq X$; $A = \{(A, B), (B, E)\}$

- **Itération 3:**

On examine l'arc (E,C) et (E,D)

$$\pi(E) + d((E,C)) = 4 + 1 = 5 < \pi(C) = 6 \Rightarrow \pi(C) = 5; A(C) = (E, C)$$

$$\pi(E) + d((E,D)) = 4 + 7 = 11 < \pi(D) = \infty \Rightarrow \pi(D) = 11; A(D) = (E, D)$$

x	A	B	C	D	E
$\pi(x)$	0	2	5	11	4

$$\pi(z) = \min\{\pi(y) / y \notin S\} = \min\{\pi(C), \pi(D)\} = \min\{5, 11\} = 5 = \pi(C) \text{ Donc } z=C$$

On pose alors: $\alpha = C$; $S = \{A, B, E, C\} \neq X$; $A = \{(A, B), (B, E), (E, C)\}$

- **Itération 4:**

On examine l'arc (C,D)

$$\pi(C) + d((C,D)) = 5 + 2 = 7 < \pi(D) = 11 \Rightarrow \pi(D) = 7; A(D) = (C, D)$$

x	A	B	C	D	E
$\pi(x)$	0	2	5	7	4

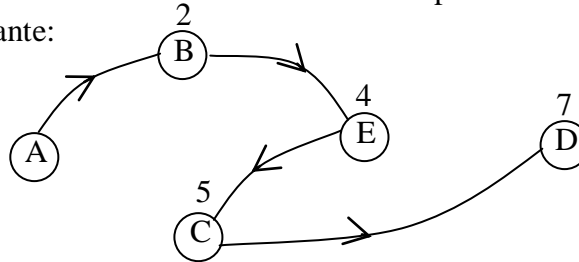
$$\pi(z) = \min\{\pi(y) / y \notin S\} = \min\{\pi(D)\} = \pi(D) = 7 \text{ Donc } z=D$$

On pose alors: $\alpha = D$; $S = \{A, B, E, C, D\} = X$; $A = \{(A, B), (B, E), (E, C), (C, D)\}$

Enfin: S=X Terminer.

x	A	B	C	D	E
$\pi(x)$	0	2	5	7	4

Les $\pi(x)$ sont les plus courtes distances. L'arborescence de plus courts chemins, issue du sommet A, dans le réseau est la suivante:



c) Algorithme général de Ford:

On applique l'algorithme général de Ford pour la recherche d'un plus court chemin sur un réseau quelconque avec $d(u) \in \mathbb{R}$.

Cet algorithme permet soit:

- De mettre en évidence un circuit absorbant si celui-ci existe.
- De déterminer une arborescence des plus courts chemins de racine s dans un réseau s'il ne contient pas de circuit absorbant.

Le principe:

Le principe de l'algorithme général de Ford consiste à améliorer une arborescence réalisable (initiale) (X, A) de racine s jusqu'à l'obtention d'une arborescence optimale des plus courts chemins, issue de s si celle-ci existe.

Enoncé:

Données: Un réseau $R=(X, U, d)$ avec $d(u) \in \mathbb{R}$.

Résultat: Arborescence des plus courtes distances A.

(0) Initialisation: Soit (X, A) une arborescence de racine s dans le réseau R et $\pi(x)$ les longueurs des chemins de s à x dans l'arborescence (X, A)

on utilise par exemple l'algorithme de Dijkstra pour obtenir l'arborescence réalisable (X, A) .

(1) Chercher un arc $u=(i, j)$ dans le réseau R, n'appartenant pas à A, tel que:

$$\delta(u) = \pi(j) - \pi(i) - d(i, j) > 0$$

- Si un tel arc n'existe pas; Terminer, (X, A) est optimale.
- Si un tel arc existe; Aller à (2)

(2) Tester Si $(X, A \cup \{u\})$ contient un circuit

- Si oui; examiner si ce circuit est absorbant (s'il l'est; terminer. Le problème n'admet pas de solution).
- Sinon; Aller à (3).

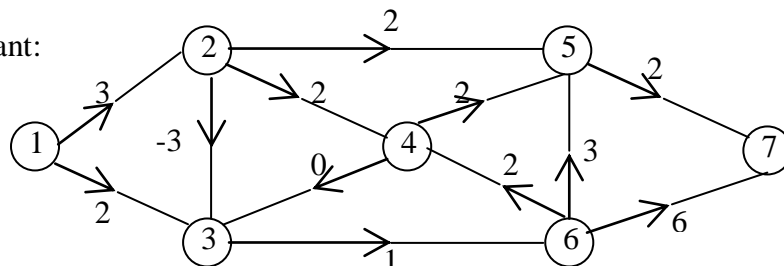
(3) Chercher un arc $v \in A$ tel que $T(v)=j=T(u)$. On pose: $A = A \cup \{u\} / \{v\}$

Soit $X' = \{j\} \cup \{\text{descendant de } j \text{ dans l'arborescence } A\}$.

On pose: $\pi(y) = \pi(y) - \delta(u), \forall y \in X'$ Aller en (1).

Application:

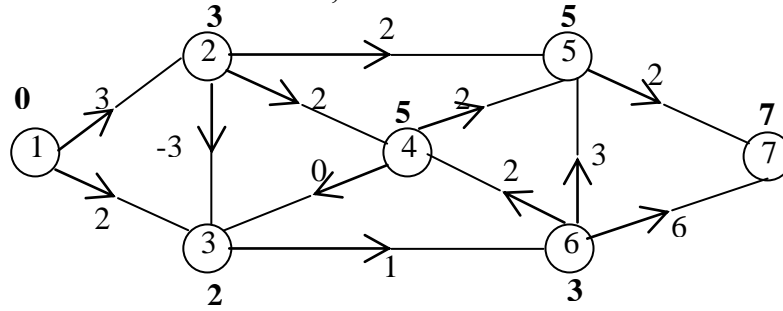
Soit le réseau $R=(X, U, d)$ suivant:



Ce réseau ne possède pas de circuit absorbant et le sommet 1 est une racine; alors ce problème admet une arborescence des plus courts chemins.

Initialisation:

Soit $A = \{(1,2), (1,3), (3,6), (6,4), (2,5), (5,7)\}$ une arborescence initiale.



- Itération 1:

Dans le réseau R, les arcs n'appartenant pas à A sont: (2,4), (2,3), (4,3), (4,5), (6,5), (6,7) tels que:

$$\delta(2,3) = \pi(3) - \pi(2) - d(2,3) = 2 - 3 + 3 = 2$$

$$\delta(2,4) = \pi(4) - \pi(2) - d(2,4) = 5 - 3 - 3 = 0$$

$$\delta(4,3) = \pi(3) - \pi(4) - d(4,3) = 2 - 5 - 0 = -3$$

$$\delta(4,5) = \pi(5) - \pi(4) - d(4,5) = 5 - 5 - 2 = -2$$

$$\delta(6,5) = \pi(5) - \pi(6) - d(6,5) = 3 - 5 - 3 = -3$$

$$\delta(6,7) = \pi(7) - \pi(6) - d(6,7) = 7 - 3 - 6 = -2$$

L'arc $u = (2,3) \notin A$ et $\delta(u) > 0$ et $A \cup \{(2,3)\}$ ne contient pas de circuit, donc l'arc $u=(2,3)$ entre dans l'arborescence.

Soit l'arc $v = (1,3) \in A, T(v) = 3$ donc l'arc (1,3) sort de l'arborescence.

Soit $X' = \{3\} \cup \{\text{descendants de sommet 3 dans l'arborescence } A\} = \{3, 4, 6\}$

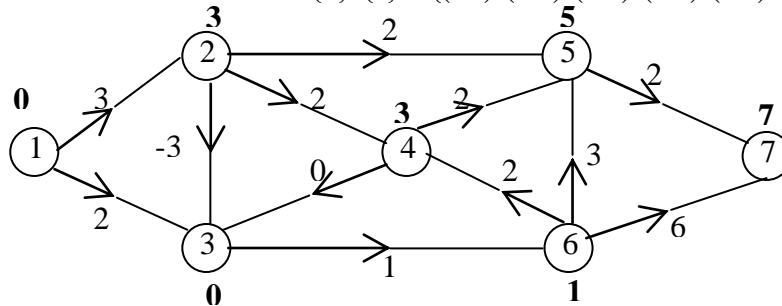
On pose:

$$\pi(3) = \pi(3) - \delta(2,3) = 2 - 2 = 0$$

$$\pi(6) = \pi(6) - \delta(2,3) = 3 - 2 = 1$$

$$\pi(4) = \pi(4) - \delta(2,3) = 5 - 2 = 3$$

On obtient ainsi la nouvelle arborescence $A = A \cup \{u\} / \{v\} = \{(1,2), (2,5), (5,7), (2,3), (3,6), (6,4)\}$



- Itération 2:

Dans le réseau R, les arcs n'appartenant pas à la nouvelle arborescence A sont: (2,4), (1,3), (4,3), (4,5), (6,5), (6,7) tels que:

$$\delta(1,3) = \pi(3) - \pi(1) - d(1,3) = 0 - 0 - 3 = -2$$

$$\delta(2,4) = \pi(4) - \pi(2) - d(2,4) = 3 - 3 - 2 = -2$$

$$\delta(4,3) = \pi(3) - \pi(4) - d(4,3) = 0 - 3 - 0 = -3$$

$$\delta(4,5) = \pi(5) - \pi(4) - d(4,5) = 5 - 3 - 2 = 0$$

$$\delta(6,5) = \pi(5) - \pi(6) - d(6,5) = 5 - 1 - 3 = 1$$

$$\delta(6,7) = \pi(7) - \pi(6) - d(6,7) = 7 - 1 - 6 = 0$$

L'arc $u = (6,5) \notin A$ et $\delta(u) > 0$ et $A \cup \{(6,5)\}$ ne contient pas de circuit, donc l'arc $u=(6,5)$ entre dans l'arborescence.

Soit l'arc $v = (2,5) \in A, T(v) = 5$ donc l'arc $(2,5)$ sort de l'arborescence.

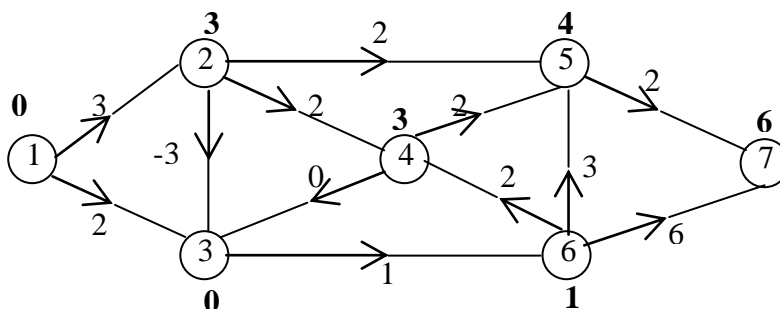
Soit $X' = \{5\} \cup \{\text{descendants de sommet 5 dans l'arborescence } A\} = \{5,7\}$

On pose:

$$\pi(5) = \pi(5) - \delta(6,5) = 5 - 1 = 4$$

$$\pi(7) = \pi(7) - \delta(6,5) = 7 - 1 = 6$$

On obtient ainsi la nouvelle arborescence $A = A \cup \{u\} / \{v\} = \{(1,2), (6,5), (5,7), (2,3), (3,6), (6,4)\}$



- Itération 3:

Dans le réseau R, les arcs n'appartenant pas à la nouvelle arborescence A sont: $(2,4), (1,3), (4,3), (4,5), (2,5), (6,7)$ tels que:

$$\delta(1,3) = \pi(3) - \pi(1) - d(1,3) = 0 - 0 - 2 = -2$$

$$\delta(2,4) = \pi(4) - \pi(2) - d(2,4) = 3 - 3 - 2 = -2$$

$$\delta(4,3) = \pi(3) - \pi(4) - d(4,3) = 0 - 3 - 0 = -3$$

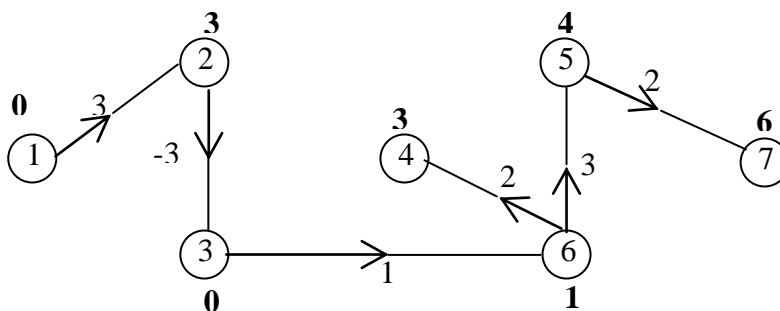
$$\delta(4,5) = \pi(5) - \pi(4) - d(4,5) = 4 - 3 - 2 = -1$$

$$\delta(2,5) = \pi(5) - \pi(2) - d(2,5) = 4 - 3 - 2 = -1$$

$$\delta(6,7) = \pi(7) - \pi(6) - d(6,7) = 6 - 1 - 6 = -1$$

Il n'existe aucun arc u n'appartient pas à A et $\delta(u) > 0$. D'où l'arborescence déjà trouvée est optimale.

L'arborescence optimale est $A = \{(1,2), (6,5), (5,7), (2,3), (3,6), (6,4)\}$



Remarque:

- ✓ La recherche d'un plus long chemin sur le réseau $R=(X,U,d)$ revient à rechercher un plus court chemin sur $R=(X,U,-d)$.
- ✓ On peut utiliser l'algorithme de Dijkstra.

CHAPITRE 4: LE PROBLÈME DU FLOT MAXIMUM

Le problème du flot maximum peut correspondre à un problème d'acheminement de tonnages disponibles sur des bateaux, des camions, des wagons ou à des canalisations, à des voies de transmission ...etc, vers une destination. Par exemple, l'alimentation journalière d'une ville en gaz peut être considérée comme un problème de flot maximum, si on s'intéresse à la quantité maximale de gaz que cette ville peut recevoir.

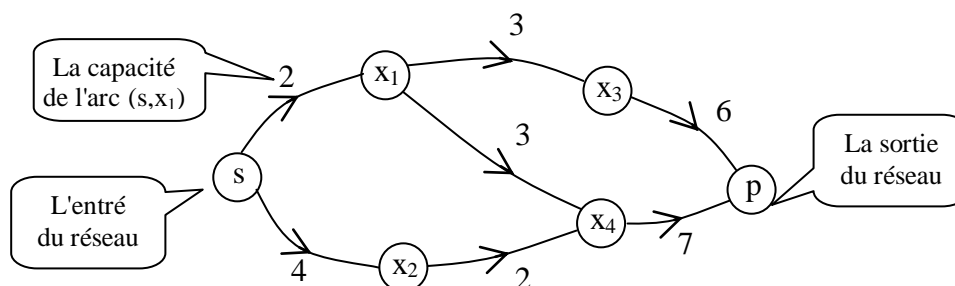
1- Définitions:

- **Réseau de transport:**

Un réseau de transport est un graphe sans boucles, où chaque arc est valué par un nombre positif $c(u)$ appelé capacité de l'arc u . Ce réseau comporte un sommet sans prédécesseurs appelé "l'entrée du réseau" ou "la source" et autre sommet sans successeurs appelé "la sortie de réseau" ou "le puits".

On note un réseau par $R=(X, U, C)$.

Exemple:



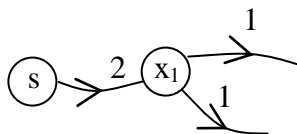
- **Flot:**

Un flot "f" dans un réseau de transport, associe à chaque arc u une quantité $f(u)$ qui représente la quantité de flux qui passe par cet arc en provenance de la source vers le puits.

Remarque:

Un flot est conservatif, s'il obéit à la règle de Kirchoff aux nœuds (aux sommets) suivantes: la somme des quantités de flux sur les arcs entrant dans un sommet doit être égale à la somme des quantités de flux sur les arcs sortant de ce même sommet.

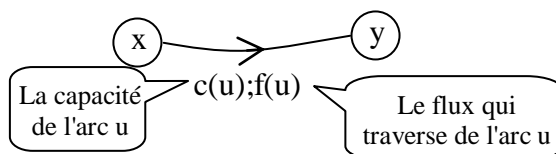
Exemple:



Dans le graphe, la quantité de flux rentrant dans x_1 est égale à la somme des quantités de flux sortant de x_1 . La quantité de flot a pour valeur 2. La loi de Kirchoff est vérifiée au sommet x_1 .

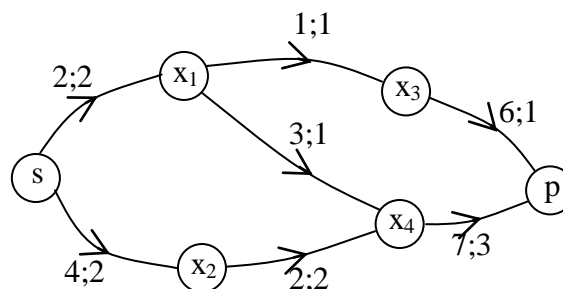
- **Flot compatible:**

Un flot est compatible dans un réseau si pour tout arc $u=(x,y)$, $0 \leq f(u) \leq c(u)$, autrement dit pour chaque arc u , le flux qui le traverse ne dépasse pas sa capacité. On retiendra dans ce qui suit la représentation suivante:



Exemple:

Soit le réseau $R=(X,U,C)$ suivant:



Dans le réseau R , le flot qui traverse chaque arc ne dépasse pas sa capacité, alors ce flot est compatible.

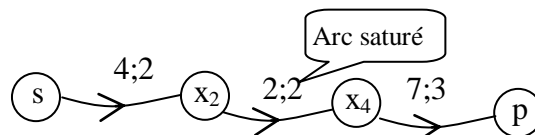
• **Flot complet:**

Un flot est complet si pour tout chemin allant de la source au puits il y'a au moins un arc saturé, c'est-à-dire: le flux qui le traverse est égal à sa capacité ($f(u)=c(u)$).

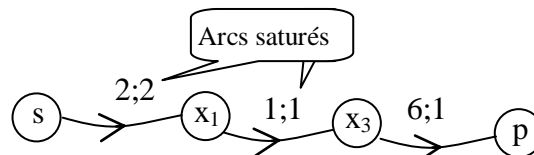
Exemple:

Dans la figure précédente, on a 3 chemins qui mènent de s à p pour lesquels on a au moins un arc saturé. Le flot est complet.

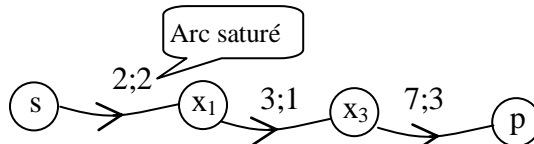
Premier chemin:



Deuxième chemin:



Troisième chemin:



2- Le problème de la recherche du flot maximum:

Le problème de flot maximum consiste à trouver la quantité maximum de flot à acheminer de la source s vers le puits p , en tenant compte des capacités de transport et de la quantité disponible en s .

Algorithme de Ford et Fulkerson pour la recherche d'un flot maximum:

L'algorithme le plus connu pour résoudre ce problème est celui de Ford et Fulkerson.

Le principe:

L'idée de l'algorithme de Ford et Fulkerson est de faire passer un flot compatible dans le réseau, le plus évident est le flot nul, puis l'améliorer jusqu'à ce qu'on obtienne un flot complet.

Une chaîne pour laquelle le flot peut être augmenté est une chaîne dont les arcs dans le sens direct n'ont pas atteint leur limite et les arcs dans le sens indirect ont un flux non nul qui les traverse.

Autrement dit: une chaîne C est dite augmentante si:

- Pour tout arc u direct de C , c'est-à-dire $u \in C^+$: $f(u) < c(u)$
- Pour tout arc u indirect de C , c'est-à-dire $u \in C^-$: $f(u) > 0$

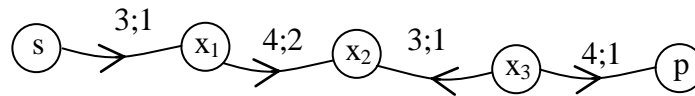
Le flot sur cette chaîne C peut être augmenté de la valeur suivante:

$$\varepsilon = \text{Minimum entre } \{c(u) - f(u)/u \in C^+\} \text{ et } \{f(u)/u \in C^-\}$$

Pour améliorer le flot, on ajoute ε au flot des arcs C^+ , c'est-à-dire les arcs directs dans la chaîne, et on retranche au flot des arcs de C^- , c'est-à-dire, les arcs indirects, dans la chaîne.

Exemple:

Voici une chaîne C reliant les sommets s et p prise d'un réseau de transport dont le flot peut être augmenté:



Dans la chaîne on a:

- Les arcs dans le sens direct $C^+ = \{(s, x_1), (x_1, x_2), (x_3, p)\}$ n'ont pas atteint leur limite: $f(u) < c(u)$.
- Les arcs dans le sens indirect $C^- = \{(x_3, x_2)\}$ ont un flux non nul ; $f(u) > 0$.

D'où la chaîne C est augmentante.

Le flot sur cette chaîne peut être augmenté de la valeur suivante:

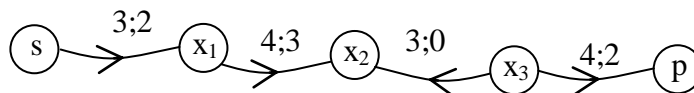
$$\varepsilon = \min\{[c(u) - f(u)/u \in C^+]; [f(u)/u \in C^-]\}$$

$$= \min[3-1; 4-2; 4-1; 1] = 1$$

On augmentera donc le flot de cette chaîne de 1, ce qui signifie:

- Augmenter de 1 le flux entre s et x_1 .
- Augmenter de 1 le flux entre x_1 et x_2 .
- Diminuer de 1 le flux entre x_3 et x_2 .
- Augmenter de 1 le flux entre x_3 et p .

On obtient alors le nouveau flot sur la chaîne:



On remarque que pour les arcs en sens inverse, améliorer le flot signifie réduire le flux les traversant. Entre x_3 et x_2 , le flux est réduit d'une unité pour permettre l'arrivée d'une unité de flux sur x_2 par x_1 en augmentant le flux entre x_1 et x_2 d'une unité et ceci en conservant la loi de Kirchhoff au sommet x_2 .

Enoncé:

Données: un 1-graphe valué $G=(X,U,c)$; f un flot maximum.

Résultat: un flot f complet.

(0) Initialisation: Marquer un sommet s et poser: $C^+ = \emptyset$; $C^- = \emptyset$; $f^k = 0$; $A = \{s\}$; $k := 0$

(1) Soit A l'ensemble des sommets marqués et soit x un sommet de A :

- Marquer le sommet y successeur de x tel que $f(x, y) < c(x, y)$
On pose: $C^+ := C^+ \cup \{(x, y)\}$; $A := A \cup \{y\}$
- Marquer le sommet y prédécesseur de x tel que $f(x, y) > 0$
On pose: $C^- := C^- \cup \{(x, y)\}$; $A := A \cup \{y\}$

Quand on ne peut plus marquer, deux cas se présentent:

- 1- p est marqué aller en (2)
- 2- p n'est pas marqué, terminé le flot est maximum.

(2) On a obtenu une chaîne augmentante $C = C^+ \cup C^-$ de s à p .

Pour améliorer le flot on calcule:

- $\varepsilon_1 = \min[c(u) - f(u); u \in C^+]$
- $\varepsilon_2 = \min[f(u); u \in C^-]$

D'où $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$

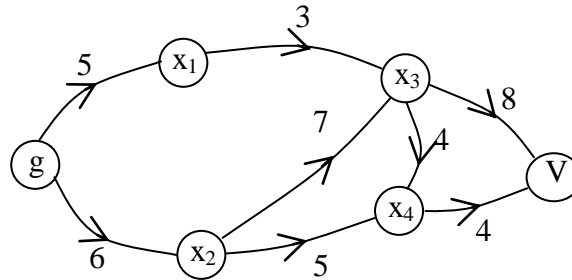
On définit le nouveau flot:

$$f^{k+1}(u) = \begin{cases} f^k(u) + \varepsilon & \text{pour } u \in C^+ \\ f^k(u) - \varepsilon & \text{pour } u \in C^- \\ f^k(u) & \text{pour } u \notin C \end{cases}$$

Effacer les marques sauf en s , et aller en (1).

Application:

Une usine à gaz alimente une ville V par l'intermédiaire du réseau de distribution ci-dessous. Les nombres associés aux arcs représentent les capacités de transport.

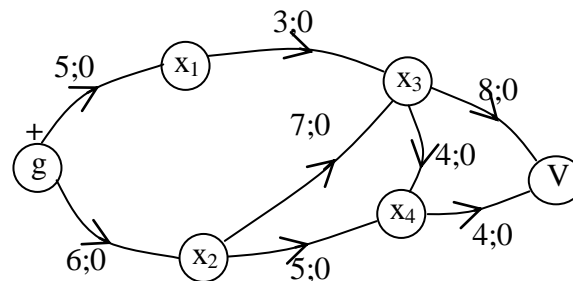


On voudrait connaître la quantité maximale que peut écouler l'usine. Ce qui revient à chercher un flot maximum sur le réseau.

- Initialisation:

On marque le sommet g (entré du réseau R) par le signe +.

On pose : $A = \{g\}$; $C^+ \cup C^- = \emptyset$ et $f^k = 0$; un flot défini sur le réseau R, $k=0$



Le réseau R après défini le flot f^0

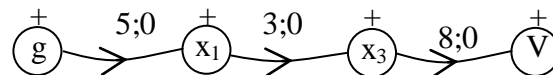
- Itération 1:

Dans le réseau R, on suit la procédure de marquage suivante:

- On marque le sommet x_1 d'un +, car il est successeur de g et $f^0(g, x_1) = 0 < c(g, x_1) = 5$
On pose: $C^+ = C^+ \cup \{(g, x_1)\} = \{(g, x_1)\}$; $A = A \cup \{x_1\} = \{g, x_1\}$.
- On marque le sommet x_3 d'un +, car il est successeur de x_1 et $f^0(x_1, x_3) = 0 < c(x_1, x_3) = 3$
On pose: $C^+ = C^+ \cup \{(x_1, x_3)\} = \{(g, x_1), (x_1, x_3)\}$; $A = A \cup \{x_3\} = \{g, x_1, x_3\}$.
- On marque le sommet V d'un +, car il est successeur de x_3 et $f^0(x_3, V) = 0 < c(x_3, V) = 8$
On pose: $C^+ = C^+ \cup \{(x_3, V)\} = \{(g, x_1), (x_1, x_3), (x_3, V)\}$; $A = A \cup \{V\} = \{g, x_1, x_3, V\}$.

Le sommet V était marqué, la procédure s'arrête. On obtient donc la chaîne augmentante

$C = C^+ \cup C^- = C^+ = \{(g, x_1), (x_1, x_3), (x_3, V)\}$ reliant le sommet g et V

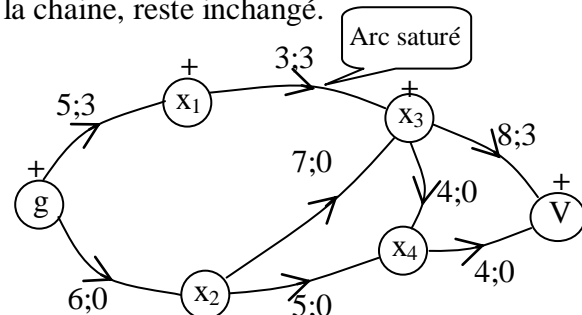


On calcule: $\varepsilon_1 = \min[c(u) - f(u); u \in C^+]$

$$= \min[c(g, x_1) - f^0(g, x_1); c(x_1, x_3) - f^0(x_1, x_3); c(x_3, V) - f^0(x_3, V)]$$

$$= \min[5 - 0; 3 - 0; 8 - 0] = 3$$

On améliore ainsi le flot f^0 pour obtenir un nouveau flot f^1 , en ajoutant la quantité ε_1 au flot des arcs de C^+ . Le flux des arcs n'appartenant pas à la chaîne, reste inchangé.



Le réseau R après avoir défini le nouveau flot f^1

On efface les marques sauf en g.

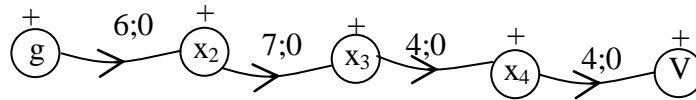
- **Itération 2:**

Dans le réseau R, on suit la procédure de marquage suivante:

- On marque le sommet x_1 d'un +, car il est successeur de g et $f^1(g, x_1) = 3 < c(g, x_1) = 5$
On pose: $C^+ = C^+ \cup \{(g, x_1)\} = \{(g, x_1)\}$; $A = A \cup \{x_1\} = \{g, x_1\}$.
Le sommet x_3 est le successeur de x_1 , mais l'arc (x_1, x_3) est saturé $f^1(x_1, x_3) = c(x_1, x_3) = 3$
Donc, on ne peut pas marquer x_3 (on abandonne ce marquage).
- On marque le sommet x_2 d'un +, car il est successeur de g et $f^1(g, x_2) = 0 < c(g, x_2) = 6$
On pose: $C^+ = C^+ \cup \{(g, x_2)\} = \{(g, x_2)\}$; $A = A \cup \{x_2\} = \{g, x_2\}$.
- On marque le sommet x_3 d'un +, car il est successeur de x_2 et $f^1(x_2, x_3) = 0 < c(x_2, x_3) = 7$
On pose: $C^+ = C^+ \cup \{(x_2, x_3)\} = \{(g, x_2), (x_2, x_3)\}$; $A = A \cup \{x_3\} = \{g, x_2, x_3\}$.
- On marque le sommet x_4 d'un +, car il est successeur de x_3 et $f^1(x_3, x_4) = 0 < c(x_3, x_4) = 4$
On pose: $C^+ = C^+ \cup \{(x_3, x_4)\} = \{(g, x_2), (x_2, x_3), (x_3, x_4)\}$; $A = A \cup \{x_4\} = \{g, x_2, x_3, x_4\}$.
- On marque le sommet V d'un +, car il est successeur de x_4 et $f^1(x_4, V) = 0 < c(x_4, V) = 4$
On pose: $C^+ = C^+ \cup \{(x_4, V)\} = \{(g, x_2), (x_2, x_3), (x_3, x_4), (x_4, V)\}$; $A = A \cup \{V\} = \{g, x_2, x_3, x_4, V\}$.

Le sommet V est marqué, on arrête le marquage. La chaîne obtenue est augmentante

$C = C^+ \cup C^- = C^+ = \{(g, x_2), (x_2, x_3), (x_3, x_4), (x_4, V)\}$, et relie les deux sommets g et V

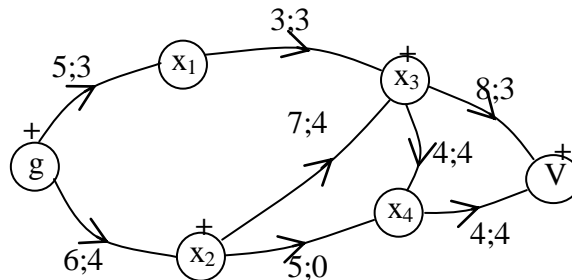


On calcule: $\varepsilon_1 = \min[c(u) - f(u); u \in C^+]$

$$= \min \left[\begin{array}{c} c(g, x_2) - f^1(g, x_2); c(x_2, x_3) - f^1(x_2, x_3); c(x_3, x_4) - f^1(x_3, x_4); \\ c(x_4, V) - f^1(x_4, V) \end{array} \right]$$

$$= \min[6 - 0; 7 - 0; 4 - 0; 4 - 0] = 4$$

On améliore ainsi le flot f^1 pour obtenir un nouveau flot f^2 , en ajoutant la quantité ε_1 au flot des arcs de C^+ . Le flux des arcs n'appartenant pas à la chaîne, reste inchangé.



Le réseau R après défini le flot f^2

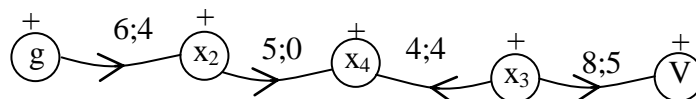
On efface les marques sauf en g.

- **Itération 3:**

Dans le réseau R, on suit la procédure de marquage suivante:

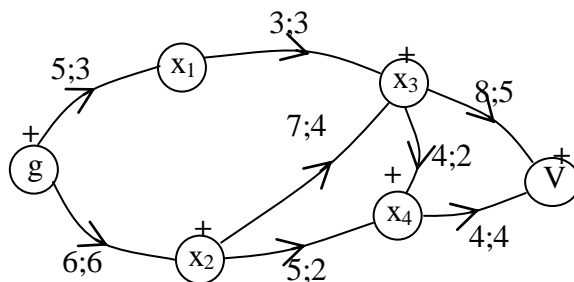
- On marque le sommet x_2 d'un +, car il est successeur de g et $f^2(g, x_2) = 4 < c(g, x_2) = 6$
On pose: $C^+ = C^+ \cup \{(g, x_2)\} = \{(g, x_2)\}$; $A = A \cup \{x_2\} = \{g, x_2\}$.
- On marque le sommet x_4 d'un +, car il est successeur de x_2 et $f^2(x_2, x_4) = 0 < c(x_2, x_4) = 5$
On pose: $C^+ = C^+ \cup \{(x_2, x_4)\} = \{(g, x_2), (x_2, x_4)\}$; $A = A \cup \{x_4\} = \{g, x_2, x_4\}$.
- On marque le sommet x_3 d'un +, car il est prédécesseur de x_4 et $f^2(x_3, x_4) = 4 > 0$
On pose: $C^- = C^- \cup \{(x_3, x_4)\} = \{(x_3, x_4)\}$; $A = A \cup \{x_3\} = \{g, x_2, x_4, x_3\}$.
- On marque le sommet V d'un +, car il est successeur de x_3 et $f^2(x_3, V) = 5 < c(x_3, V) = 8$
On pose:
 $C^+ = C^+ \cup \{(x_3, V)\} = \{(g, x_2), (x_2, x_4), (x_3, V)\}$; $A = A \cup \{V\} = \{g, x_2, x_4, x_3, V\}$.

Le sommet V est marqué, on arrête le marquage. La chaîne obtenue est augmentante $C = C^+ \cup C^- = \{(g, x_2), (x_2, x_4), (x_3, x_4), (x_3, V)\}$, et relie les deux sommets g et V



On calcule: $\varepsilon_1 = \min[c(u) - f(u); u \in C^+]$
 $= \min[c(g, x_2) - f^2(g, x_2); c(x_2, x_4) - f^2(x_2, x_4); c(x_3, V) - f^2(x_3, V)]$
 $= \min[6 - 4; 5 - 0; 8 - 5] = 2$
 $\varepsilon_2 = \min[f(u); u \in C^-]$
 $= \min[f^2(x_3, x_4)] = 4$
 $\varepsilon = \min[\varepsilon_1, \varepsilon_2] = \min[2, 4] = 2$

On améliore ainsi le flot f^2 pour obtenir un nouveau flot f^3 , en ajoutant la quantité ε au flot des arcs de C^+ et retranchant la quantité ε au flot des arcs C^- . Le flux des arcs n'appartenant pas à la chaîne, reste inchangé.



Le réseau R après défini le flot f^3

On efface les marques sauf en g.

Itération 4:

Dans le réseau R, on ne peut pas marquer le sommet V. Donc le flot obtenu est maximum, on le représente comme suit:

Arcs	(g,x ₁)	(g,x ₂)	(x ₁ ,x ₃)	(x ₂ ,x ₃)	(x ₂ ,x ₄)	(x ₃ ,x ₄)	(x ₃ ,V)	(x ₄ ,V)
flux	3	6	3	4	2	2	5	4

La valeur du flot maximum est égale à celle des flux sortant de la source g, ou la somme des valeurs des flux entrant au puits p.

C'est-à-dire: $f_{max} = \sum(f(s, x)/x \in \Gamma^+(s)) = \sum(f(x, p)/x \in \Gamma^-(p))$

La production maximale que peut écouler l'usine g vers la ville V:

$f_{max} = f^3(x_3, V) + f^3(x_4, V) = 5 + 4 = 9$ ou $f_{max} = f^3(g, x_1) + f^3(g, x_2) = 3 + 6 = 9$

Remarque 1:

Pour accélérer le processus de résolution de problème de recherche de flot maximum, on démarre avec un flot au jugé, et on essaie de le rendre complet.

Un flot au jugé, consiste à envoyer une matière à partir du sommet s, et de la distribuer sur le réseau tout en respectant la conservation de la matière en chaque sommet. On applique ensuite l'algorithme de Ford et Fulkerson avec comme flot de départ, le flot complet obtenu.

Remarque 2:

Lors de l'application de l'algorithme de Ford et Fulkerson; on peut déterminer dans un premier temps un flot complet, c'est-à-dire, déterminer toutes les chaînes augmentantes qui sont des chemins, puis dans un second temps, chercher toutes celles qui permettront de rendre le flot complet maximum.

CHAPITRE 5: LE PROBLÈME D'ORDONNANCEMENT

La gestion d'un projet composé de plusieurs tâches, présente de grands problèmes quant à l'établissement d'un calendrier du déroulement et du contrôle de leur exécution. C'est ainsi que sont apparus les problèmes d'ordonnancement dans la planification des projets, et ce dans le but de gagner du temps dans leur réalisation (minimiser la durée de réalisation d'un projet) compte tenu des contraintes d'antériorité reliant les différentes tâches. C'est-à-dire, une tâche ne peut commencer que si une, ou plusieurs prennent fin.

Résoudre un problème d'ordonnancement, c'est d'abord donner l'**ordre**, dans lequel doivent être exécutées les tâches, de façon à **minimiser la durée d'exécution totale du projet**, tout en satisfaisant les conditions d'antériorité. A partir de cette planification, nous constatons qu'un retard dans la réalisation de certaines tâches n'influencera pas sur la durée du projet, alors que d'autres tâches dites **critiques** retardent le projet au moindre retard local.

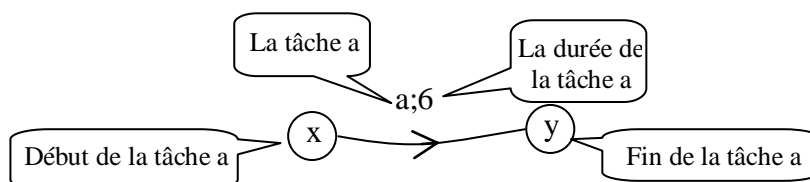
Deux grandes méthodes sont utilisées pour résoudre un problème d'ordonnancement de projet, on a la méthode américaine CPM (Critical Path Method) avec sa variation PERT (Program Evaluation and Review Technique) et la méthode française MPM (méthode des potentiels). Nous nous intéresserons à la méthode de PERT.

1- La représentation du réseau PERT :

La méthode PERT consiste à représenter un problème d'ordonnancement de projet par un graphe dit **réseau PERT** ou **diagramme événement-tâche**, qu'on note $R=(X,U,d)$ où:

- Un arc correspond à une tâche.
- La valeur d'un arc u représente la durée d'une tâche $d(u)$.
- Un sommet est un **événement** signifiant le début ou la fin d'une ou plusieurs tâches.

Exemple:



Remarque:

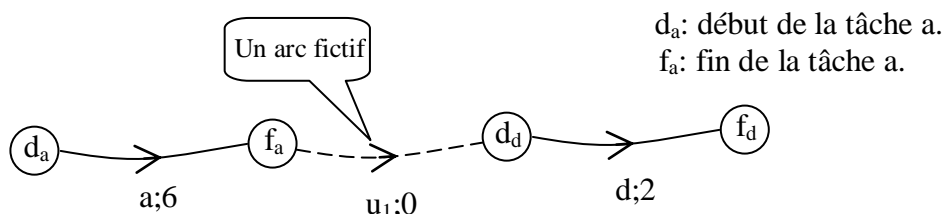
Soit $R=(X,U,d)$ un réseau PERT.

La succession de deux tâches consistera en premier lieu à ajouter une tâche fictive de durée nulle pour indiquer que l'événement fin de la première tâche coïncide avec l'événement début de la tâche suivante, on distinguera les deux cas suivants:

- **Premiers cas:**

Soit a et d deux tâches du réseau R , telles que la tâche a précède la tâche d .

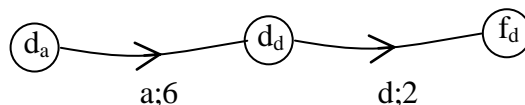
La représentation des deux tâches par un arc fictif:



L'arc u_1 correspond à la tâche fictive de durée nulle.

La représentation des deux tâches sans arc fictif:

Pour simplifier le graphe, on peut supprimer la tâche fictive, et obtenir ainsi la représentation ci-dessous.

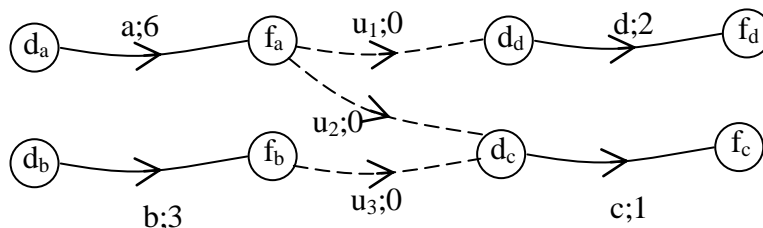


• Deuxième cas:

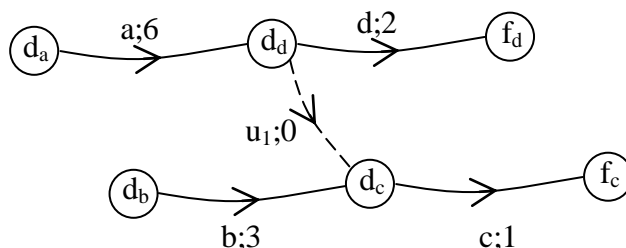
Soit a, b, c et d trois tâches de R telle que:

- La tâche a précède les tâches d et c.
- La tâche b précède la tâche c.

La première représentation des tâches est comme suit:



Les tâches fictives u_1 et u_3 peuvent être supprimées alors que u_2 est nécessaire; elle signifie que a procède c en même temps que d, tout en évitant la confusion quant à l'antériorité unique de a par rapport à d, on obtient ainsi la représentation suivante:



Remarque:

On rajoute deux sommets D et F au réseau de PERT pour représenter respectivement le début et la fin du projet. La représentation graphique est ordonnée par niveaux des sommets (des événements).

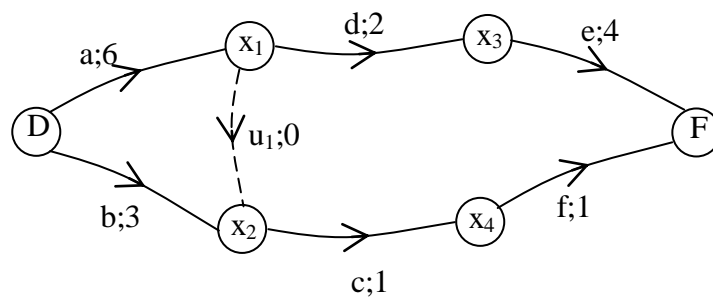
Exemple:

Un projet est composé de 6 tâches dont les durées de réalisation et les contraintes d'antériorité (de précedence) sont données dans le tableau suivant:

Tâches	Tâches précédentes	Durées des tâches
a	-	6
b	-	3
c	b-a	1
d	a	2
e	d	4
f	c	1

Il s'agit de trouver un calendrier de déroulement des tâches, c'est-à-dire déterminer les dates de démarrage de chaque tâche, qui minimisant le temps total de réalisation du projet et d'indiquer les tâches critiques.

La représentation graphique du problème est le réseau PERT $R=(X,U,d)$ suivant:



Les tâches a et b ne sont précédées par aucune tâche, alors le début de ces tâches coïncident avec le début de projet (D), de même, les tâches e et f ne sont suivies par aucune tâche, alors la fin de ces tâches coïncident avec la fin du projet (F).

2- La détermination du calendrier des dates au plus tôt et des dates au plus tard :

2-1 Le calendrier des dates au plus tôt des événements :

Une fois qu'on aura établi le réseau de PERT, on pourra déterminer pour chaque événement la date à laquelle il peut au plus tôt se réaliser, autrement dit, la date de début au plus tôt de chaque tâche.

On considère la date de début de l'événement D (début du projet) égale à 0.

Un événement x ne se réalise que si tous les événements le précédant se sont réalisés.

On calcule alors la longueur maximale du chemin aboutissant à x.

La date au plus tôt d'un événement x notée t_x se calcule comme suit

$$t_x = [t_y + d(x,y)/y \in \Gamma_G^-(x)]; \text{ Le max étant pris sur les prédécesseurs de x.}$$

Exemple:

Dans le réseau de PERT précédent, les dates au plus tôt des événements se calcule comme suit:

- $t_D = 0$
- $t_{x_1} = t_D + d(a) = 0 + 6 = 6$
- $t_{x_2} = \max[t_D + d(b); t_{x_1} + d(u_1)] = \max[0 + 3; 6 + 0] = 6$
- $t_{x_3} = t_{x_1} + d(d) = 6 + 2 = 8$
- $t_{x_4} = t_{x_2} + d(c) = 6 + 1 = 7$
- $t_F = \max[t_{x_3} + d(e); t_{x_4} + d(f)] = \max[8 + 4; 7 + 1] = 12$

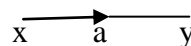
Remarque:

Pour le sommet F fin du projet, t_F correspond à la durée minimale de réalisation du projet.

2-2 Le calendrier des dates de début au plus tôt des tâches :

La date de début au plus tôt de la tâche a égale à la date au plus tôt de l'événement d'où elle est issue, on la note T_a ;

$T_a = t_x$ si la tâche a est issue de l'événement x.



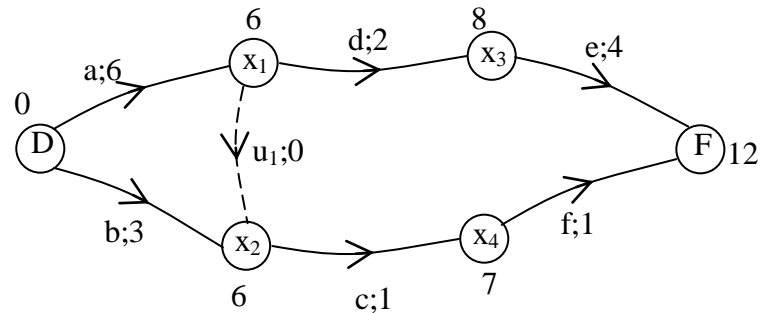
Exemple:

Dans le réseau de PERT précédent, les dates de début au plus tôt des tâches sont:

$$T_a = t_D = 0; T_b = t_D = 0; T_c = t_{x_2} = 6; T_d = t_{x_1} = 6; T_e = t_{x_3} = 8; T_f = t_{x_4} = 7;$$

Pour chaque événement, on a obtenu la date à laquelle il peut se réaliser au plus tôt, ce qui permet la planification des tâches une plus petite durée possible du projet, soit 12 jours.

On représente sur le réseau R les dates au plus tôt des événements.



2-3 Le calendrier des dates au plus tard des événements :

De la même manière on va déterminer la date à laquelle un événement peut au plus tard se délaier prévus. Pour cela, connaissant la date de fin du projet on pourra déterminer la date au plus tard à laquelle les événements peuvent se réaliser.

Dans l'exemple, l'événement F se réalise à la date 12, cela signifie que la tâche e de durée 4 doit démarrer au plus tard à la date $8=12-4$.

Si un événement a plusieurs successeurs, on calculera le minimum de la différence entre la date au plus tard de ces événements moins la durée des tâches qui y aboutissent. On note la date au plus tard d'un événement par t_x^* .

$t_F^* = t_F$ pour le sommet fin du projet

$t_x^* = \min[t_y^* - d(x, y) / y \in \Gamma_G^+(x)]$; Le min étant pris sur les successeurs y de x.

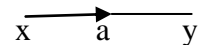
Exemple:

Dans le réseau de PERT précédent, les dates au plus tard des événements se calculent comme suit:

- $t_F^* = t_F = 12$
- $t_{x3}^* = t_F - d(e) = 12 - 4 = 8$
- $t_{x4}^* = t_F - d(f) = 12 - 1 = 11$
- $t_{x2}^* = t_{x4}^* - d(c) = 11 - 1 = 10$
- $t_{x1}^* = \min[t_{x2}^* - d(u_1); t_{x3}^* - d(d)] = \min[10 - 0; 8 - 2] = 6$
- $t_D^* = \min[t_{x1}^* - d(a); t_{x2}^* - d(b)] = \min[6 - 6; 10 - 3] = 0$

2-4 Le calendrier des dates de début au plus tard des tâches :

La date de début au plus tard de la tâche a est égale à la date au plus tard de l'événement auquel elle aboutit, diminuée de la durée de la tâche. On la note T_a^* tel que: $T_a^* = t_y^* - d(x, y)$.

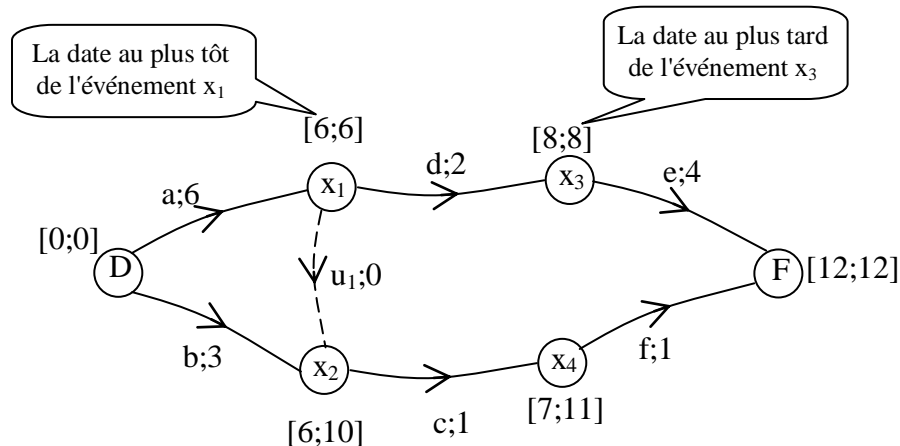


Exemple:

Dans le réseau de PERT précédent, les dates de début au plus tard des tâches sont calculées comme suit:

- $T_f^* = t_F - d(f) = 12 - 1 = 11$.
- $T_e^* = t_F - d(e) = 12 - 4 = 8$.
- $T_d^* = t_{x3}^* - d(d) = 8 - 2 = 6$.
- $T_c^* = t_{x4}^* - d(c) = 11 - 1 = 10$.
- $T_b^* = t_{x2}^* - d(b) = 10 - 3 = 7$.
- $T_a^* = t_{x1}^* - d(a) = 6 - 6 = 0$.

On détermine sur le réseau R les dates au plus tard des événements.



3- Analyse et identification des tâches critiques:

Une fois la date au plus tard et la date au plus tôt de chaque événement calculé, on peut analyser la situation. La date au plus tôt fournit la date planifiée pour chaque événement et les dates au plus tard indiquent de combien un événement peut être retardé sans retarder le projet, ceci permet d'identifier les tâches critiques.

Une tâche i correspondant à l'arc (x,y) peut voir sa durée augmentée d'un délai:

$\delta(i) = t_y^* - t_x - d(i) = T_y^* - T_x$ sans que ce retard ne se répercute nécessairement sur la durée de réalisation de projet.

3-1 Tâche critique :

Une tâche dite **critique** si tout retard dans son exécution se répercute automatiquement sur la durée de réalisation du projet, en d'autres termes, une tâche est critique si sa date de début et sa date de fin sont critiques, et si la différence entre la date de début et la date de fin est égale à la durée de la tâche.

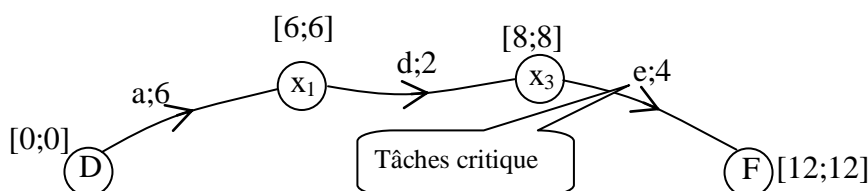
Autrement dit, une tâche est critique si $\delta(i) = 0$.

Remarque:

Il est à noter que, sur le plus long chemin de D à F, toutes les tâches sont critiques; ce chemin est dit alors chemin critique.

Exemple:

Dans le réseau R: $T_a = T_a^* = 0$; $T_d = T_d^* = 6$; $T_e = T_e^* = 8$, donc les tâches a, d et e sont critiques. Le chemin critique est donc le chemin (D, x_1, x_3, F) , de longueur 12 (longueur maximale).



3-2 Calcule des marges et de l'intervalle de flottement :

3-2-1 Intervalle de flottement:

Chaque tâche a une date de début au plus tôt, et une date de début au plus tard, le démarrage de cette tâche peut intervenir entre ces deux dates sans compromettre la date fin du projet. L'intervalle de flottement d'un événement est égal à la différence entre la date de début au plus tôt et la date de début au plus tard de la tâche.

Exemple:

Dans le réseau R:

- La tâche "f" a une date au plus tôt égale à 7, et une date au plus tard égale à 11, donc l'intervalle de flottement de l'événement début de f est [7;11].
- La tâche "c" débute au plus tôt le 6^{ème} jour, et au plus tard le 10^{ème} jour du début de projet, donc l'intervalle de flottement est de [6;10], ce qui fait que la tâche c peut débuter entre le 6^{ème} jour et le 10^{ème} jour sans retarder la date de fin du projet.

3-2-2 Les marges:

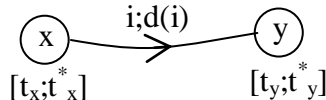
Le responsable du projet peut se pencher aussi sur le degré de liberté dont il dispose pour éventuellement augmenter la durée d'une tâche sans compromettre la durée totale du projet; on distingue trois types de marges:

☒ La marge libre:

Notée $M_l(u)$, c'est le retard maximum que l'on peut apporter au démarrage d'une tâche sans perturber la date de réalisation au plus tôt de l'événement suivant:

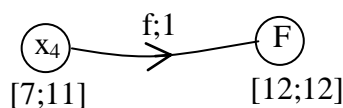
Soit une tâche i représentée par l'arc $u=(x,y)$.

$$M_l(i)=t_y-t_x-d(i)$$



Exemple:

On considère la tâche f du réseau R, la marge libre de f est $M_l(c)=12-7-1=4$



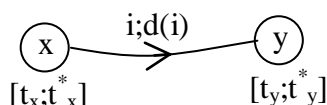
Le démarrage de la tâche f peut être retardé au maximum de 4 jours sans perturber la fin au plus tôt le projet.

☒ La marge totale:

Notée $M_t(u)$, c'est le retard maximum qu'on peut apporter au démarrage d'une tâche sans perturber la date fin du projet.

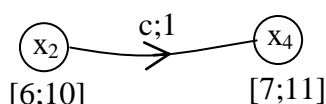
Soit une tâche i , représentée par l'arc $u=(x,y)$.

$$M_t(i)=t_y^*-t_x-d(i)=T_a^*-T_a$$



Exemple:

On considère la tâche c du réseau R, la marge totale de c est $M_t(c)=11-6-1=4$

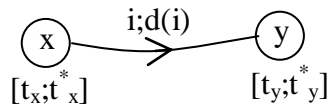


☒ **La marge certaine:**

Notée $M_c(u)$, c'est le retard maximum que l'on peut apporter au démarrage d'une tâche sans perturber la réalisation au plus tôt de l'événement suivant bien que l'événement précédent n'a été réaliser qu'à sa date limite.

Soit une tâche i , représentée par l'arc $u=(x,y)$.

$$M_c(i)=t_y-t_x^*-d(i)$$



Exemple:

On considère la tâche f du réseau R , la marge certaine de f est $M_c(f)=12-11-1=0$

