Théorie des Graphes

Nedioui Mohamed Abdelhamid

15 Avril 2022

Cours Licence 2ième année

Table de matière

Introduction

L'histoire de la théorie des graphes débuterait avec les travaux d'Euler au 18e siècle et trouve son origine dans l'étude de certains problèmes, tels que celui des ponts de Königsberg étudié par Euler en 1736, on peut dire que les premiers développements majeurs de la théorie des graphes datent du milieu du vingtième siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de König, Menger, Cayley puis de Berge et d'Erdös. Depuis cette époque, la théorie des graphes s'est largement développée et fait à présent partie du cursus standard en mathématiques de bon nombre d'universités.

La théorie des graphes trouve différentes applications dans de nombreux domaines tels que la chimie, la biologie, les réseaux de télécommunications ou encore les réseaux sociaux. Les recherches en théorie des graphes sont essentiellement menées par des informaticiens du fait de l'importance des aspects algorithmiques (recherche de solutions). Il s'agit essentiellement de modéliser des problèmes.

Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs. Les derniers travaux en théorie des graphes sont souvent effectués par des informaticiens, du fait de l'importance qu'y revêt l'aspect algorithmique.

De manière générale, un graphe permet de représenter simplement la structure, les connexions, les cheminements possibles d'un ensemble complexe comprenant un grand nombre de situations, en exprimant les relations, les d´dépendances entre ses éléments (e.g., réseau de communication, réseaux ferroviaire ou routier, arbre généalogique, diagramme de succession de tâches en gestion de projet, ...).

En plus de son existence purement mathématique, le graphe est aussi une structure de données puissante pour l'informatique.

Ce cours constitue une introduction à la théorie des graphes. Le contenu est composé de six chapitres. Le premier « **Définitions de base** » présente les concepts généraux. Le deuxième «**Cycles**» parle des nombres cyclomatique et cocyclomatique, la planarité des graphes et les arbres. Le troisième « **Flots** » parle plus particulièrement des réseaux de transport et introduit les méthodes fondamentales de recherche d'un flot maximum. Le quatrième « **Problèmes de cheminement** » propose des méthodes générales concernant l'énumération et l'existence de plus court chemins et d'arbres à coût

extrémum. Le cinquième « **Problèmes Hamiltonien et Eulérien** » introduit les chaines (chemins) et cycles (circuits) Hamiltonien et Eulérien ainsi que quelques méthodes fondamentales liées. Le dernier «**Coloration**» donne la définition de coloration des arcs et sommets et aborde certainement l'un des plus fameux sujets de la théorie des graphes, le théorème des "4 couleurs".

Chapitre 1

Définitions de base

1 Définition de graphe

1.1 Définition

Un **graphe** est un schéma constitué par un ensemble de **points** (appelés sommets ou nœuds) reliés entre eux par un ensemble de **lignes** ou de **flèches** (appelées arêtes ou arcs).

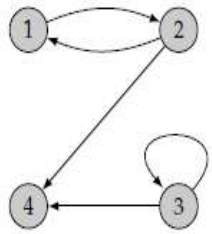
De façon plus formelle, un graphe est défini par un couple G = (S;A) tel que :

- S est un ensemble fini de sommets,
- A est un ensemble de couples de sommets $(s_i; s_j) \in S^2$.

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

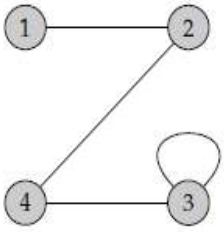
- Les liens routiers
- Les réseaux de communication
- Les circuits électriques
- Les liens entre diverses personnes.

Exemple de graphe orienté :



$G = (S, A)$ où
— $S = \{1, 2, 3, 4\}$,
— $A = \{(1, 2), (2, 1), (2, 4), (3, 4), (3, 3)\}$.

Exemple de graphe non-orienté :



$G = (S, A)$ où
— $S = \{1, 2, 3, 4\}$,
— $A = \{\{1, 2\}, \{2, 4\}, \{3, 4\}, \{3\}\}$.

1.2. Définition mathématique d'un graphe

Un **graphe** G = (X , U) est le couple constitué:

1- par un **ensemble** $X = \{x_1, x_2, \ldots, x_n\}$

2- par une **famille** $U = \{u_1, u_2, \ldots, u_n\}$ d'éléments du produit cartésien

$X \times X = \{(x , y) / x \in X , y \in X \}$

2. Ordre, orientation et multiplicité

2.1. Ordre

Selon les définitions précédentes, l'ensemble de sommets est supposé fini.

Définition :

On appelle **ordre du graphe** G = (X, U) le nombre de sommets du graphe.

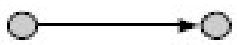
Notation :

L'ordre de G est donc le **cardinal de X** et noté |**X**|.

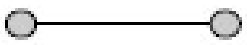
2.2. Orientation

2.2.1. Arc (arête)

On note un **arc** (**arête**) reliant un sommet x au sommet y dans un graphe G : par u =(x ,y). Chaque arc (**arête**) du graphe G relie respectivement deux sommets, le sommet de départ qui représente l'extrémité initiale de l'arc (**arête**) et le sommet d'arrivée qui représente l'extrémité terminale.



arc



arête

2.2.2. Degré

Pour un graphe non-orienté, on appelle degré d'un sommet s, noté $d(s)$ le nombre d'arêtes dont le sommet est une extrémité.

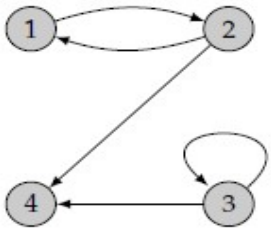
Pour un graphe orienté, soit un sommet s d'un graphe G, le nombre d'arcs de la forme (x,y) se note $d^+(s)$ et s'appelle le **demi-degré extérieur** de x. De même, le nombre d'arcs de la forme (y,x) se note $d^-(s)$ et s'appelle le **demi-degré intérieur** de x.

Le nombre $\boldsymbol{d(s) = d^+(s) + d^-(s)}$ est le **degré** du sommet s, c'est le nombre d'arcs ayant une extrémité en s (chaque boucle étant comptée deux fois).

Si tous les sommets d'un graphe ont même degré, ce graphe est un **graphe régulier.**

Exemple

Le sommet 2 est l'extrémité initiale de 2 arcs, on dit alors que le demi-degré extérieur est 2, on le note $d^+(2)= 2$ Le sommet 2 est l'extrémité terminale d'un seul arc, on dit alors que le demi-degré intérieur de sommet 2 est 1, on le note $d^-(2)= 1$



La somme du demi-degré intérieur et du demi-degré extérieur du 2 définit le degré de 2, on le note d(2)=3

2.3. Multiplicité

2.3.1. Boucle et extrémités

Un arc de la forme (x , x) est une **boucle**.

Soit un arc de la forme (x , y), x et y sont appelés les **extrémités** de l'arc :

- x est l'**extrémité initiale** de l'arc ;

- y est l'**extrémité finale** de l'arc.

2.3.2. multiplicité d'une paire x , y

La **multiplicité d'une paire x , y** est le nombre d'arcs (du graphe G) ayant x comme extrémité initiale et y comme extrémité finale.

3 Relations entre les éléments d'un graphe

3.1. Relations entre sommets

Le sommet y est un **successeur** du sommet x s'il existe un arc de la forme : (x , y).

Le sommet y est un **prédécesseur** du sommet x s'il existe un arc de la forme : (y , x)

Le sommet y est un **voisin** du sommet x s'il existe un arc de la forme (x , y) ou de la forme (y , x)

Un **sommet pendant** est un sommet qui n'a qu'un seul voisin. (autrement dit si y est un successeur ou un prédécesseur de x).

L'ensemble des successeurs du sommet x est noté: $\Gamma^{+}(x)$

L'ensemble des prédécesseurs du x est noté : $\Gamma^{-}(x)$

L'ensemble des voisins du x est noté : $\Gamma(x) = \Gamma^{+}(x) \cup \Gamma^{-}(x)$

3.2. Relations entre arcs et sommets

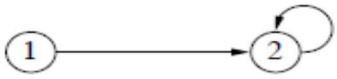
3.2.1. Arc incident à un sommet

Si (u, v) est un arc du graphe orient´e G = (V,E), on dit que l'arc (u, v) est incident aux sommets u et v ou encore que l'arc (u, v) quitte le sommet u et arrive dans le sommet v. Dans le cas non-orienté, on dit simplement que l'arête {u, v} est incidente aux sommets u et v.

Dans les deux cas, le sommet v est adjacent au sommet u. Si le graphe est non-orienté la relation adjacence est symétrique.

3.2.2. Arcs adjacents

Définition : Deux arcs sont dits adjacents s'ils ont au moins un sommet commun (une extrémité commune).



Deux arcs adjacents

3.3. Qualificatifs des graphes

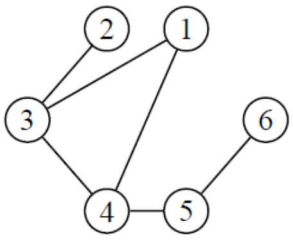
3.3.1. Sous-graphe et graphe partiel

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

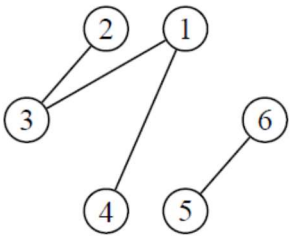
Soit A ⊂ X, alors le **sous-graphe** engendré par A est le graphe G_A dont les sommets sont les éléments de A et dont les arcs sont les arcs de G ayant leurs deux extrémités dans A.

Soit V ⊂ U alors le **graphe partiel** engendré par V est le graphe (X ,V) ayant le même ensemble X de sommets que G, et dont les arcs sont les arcs de V (on élimine de G les arcs de U - V).

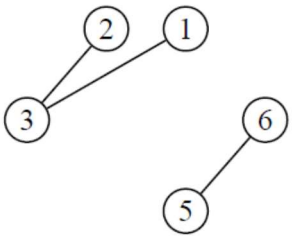
Un **sous-graphe partiel** de G est un sous-graphe d'un graphe partiel de G ou un graphe partiel d'un sous-graphe de G.



Graphe *G*



Graphe partiel de *G*

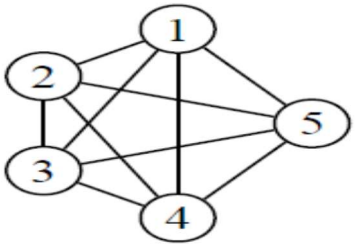


Sous-graphe de *G*

3.3.2. Graphe complet, clique

Un graphe **complet** à n sommets, noté $\mathbf{K_n}$, est un graphe non orienté sans boucle tel qu'il y a une arête entre chaque paire de sommets distincts.

Un graphe simple **complet** d'ordre **n** s'appelle une **n-clique**.



Un graphe complet à 5 sommets (k_5)

3.3.3. Graphe symétrique ou antisymétrique

Un graphe G est **symétrique** si pour tout arc (x,y), il existe un arc inverse (y,x) :

$(x,y) \in A \Rightarrow (y,x) \in A$

Un graphe G est **antisymétrique** si pour tout arc (x,y), l'arc (y,x) est absent :

$(x,y) \in A \Rightarrow (y,x) \notin A$

Attention *:*

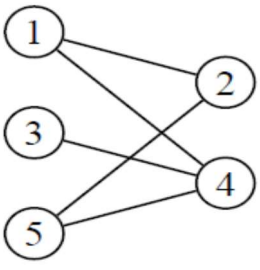
Il ne faut pas confondre graphe symétrique/anti-symétrique avec graphe non orienté / orienté.

Un graphe non orienté est fatalement symétrique, puisqu'une arête est définie par une paire de sommets, dans ordre particulier. En fait, il n'y a aucune raison de parler de graphe symétrique ou non lorsqu'on parle de graphe non orienté.

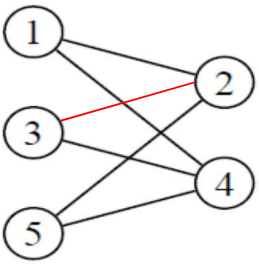
Par contre, un graphe orienté peut être symétrique ou anti-symétrique, ou ni l'un ni l'autre.

3.3.4. Graphe biparti, biparti-complet

Un graphe est biparti si l'ensemble de ses sommets peut être partitionné en deux classes et de sorte que deux sommets de la même classe ne soient jamais voisins.



Graphe biparti



Graphe biparti-complet

4 Matrices associées à un graphe

4.1 Matrice d'incidence sommet-arc

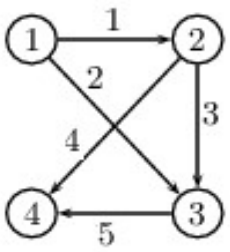
Soit G un graphe orienté qui possède **n** sommets numérotés de 1 à **n**, et **m** arcs numérotés de 1 à **m**. On appelle **matrice d'incidence** du graphe la matrice $A=(a_{i,j})$ comportant **n** lignes et **m** colonnes telle que :

- $\mathbf{a_{i,j} = +1}$, si l'arc j admet le sommet i comme origine;
- $\mathbf{a_{i,j} = -1}$, si l'arc j admet le sommet i comme arrivée;
- $\mathbf{a_{i,j} = 0}$ dans les autres cas.

Dans un graphe non-orienté, il n'y a plus de notion d'origine et d'arrivée d'une arête. On met donc +1 là où auparavant on mettait +1 ou -1, et on met 0 ailleurs.

Exemple : Voici un graphe, et la matrice d'incidence correspondante :

	1 (1,2)	2 (1,3)	3 (2,3)	4 (2,4)	5 (3,4)
1	1	1	0	0	0
2	-1	0	1	1	0
3	0	-1	-1	0	1
4	0	0	0	-1	-1



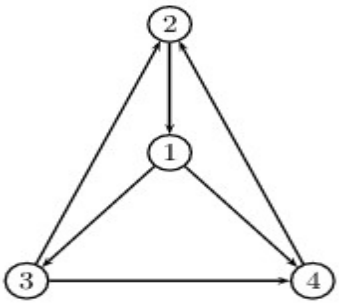
4.2 Matrice d'adjacence sommets-sommets

Soit G un graphe non-orienté qui possède n sommets numérotés de 1 à n. On appelle matrice d'adjacence du graphe la matrice $A=(a_{i,j})$ où $a_{i,j}$ est le nombre d'arêtes joignant le sommet i au sommet j.

Dans le cas G est un graphe orienté. Cette fois, le coefficient $a_{i,j}$ désigne le nombre d'arcs d'origine i et d'extrémité j.

Exemple : Voici un graphe, et la matrice d'adjacence correspondante

	1	2	3	4
1	0	0	1	1
2	1	0	0	0
3	0	1	0	1
4	0	1	0	0



5 Vocabulaire lié à la connexité

5.1. Chaîne, chemin, longueur

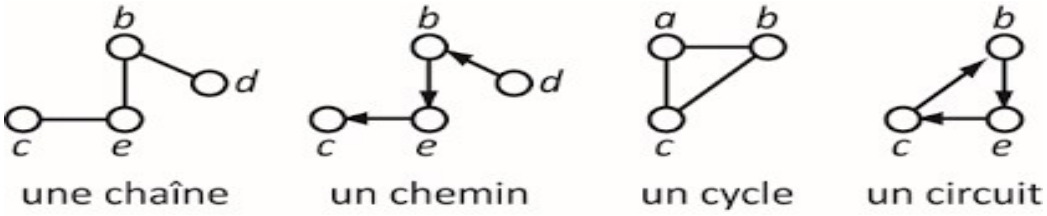
Une **chaîne** de longueur **q** > 0 est une séquence $\mu = (u_1 , u_2 , \ldots, u_q)$ d'arcs de G telle que chaque arc de la séquence est une extrémité en commun avec l'arc précédent, et l'autre extrémité en commun avec l'arc suivant. Le nombre d'arcs de la séquence est la **longueur de la chaîne**.

Une **chaîne élémentaire** est une chaîne ne rencontrant pas deux fois le même sommet.

Une **chaîne simple** est une chaîne n'utilisant pas deux fois le même arc.

Une ***chaîne*** reliant un sommet ***c*** à un sommet ***d*** est une succession d'arêtes qui permet de se rendre de *c* à *d*. Dans le cas orienté, on parle de ***chemin*** de ***u*** vers ***v***.

Un ***cycle*** dans *G* est une chaîne dont les deux extrémités coïncident. Dans le cas orienté, on parle de **circuit**.



5.2. Connexité

Un **graphe connexe** est un graphe tel que pour toute paire de sommets x et y, il existe une chaîne reliant x et y.

Un graphe est dit ***connexe*** s'il ne possède qu'une composante connexe

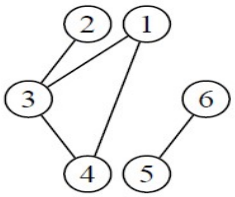
Le **nombre de connexité** du graphe est simplement le nombre de composantes connexes.

Un graphe orienté est **fortement connexe** si quels que soient *u* et *v* dans *G*, il existe un chemin reliant *u* à *v*.

Exemple : Voici un graphe connexe et graphe non connexe

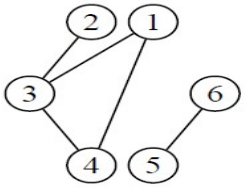
V = {1,2,3,4,5,6}

E ={{1,3},{1,4},{1,6},{2,3},{3,4},{5,6}}



V1 = {1,2,3,4} *V1* = {5,6}

E1 ={{1,3},{1,4}, {3,4},{2,3} } E2={{5,6 }}



Chapitre 2

Cycles

1. Nombres cyclomatique et cocyclomatique

    1.1. Décomposition des cycles et des cocycles en sommes élémentaires

    1.2. Lemme des arcs colorés (Minty 1960)

    1.3. Base de cycles et base de cocycles

2. Planarité

    1.1. Graphe Planaire

    2.2. Formule d'Euler

1 Nombres cyclomatique et cocyclomatique

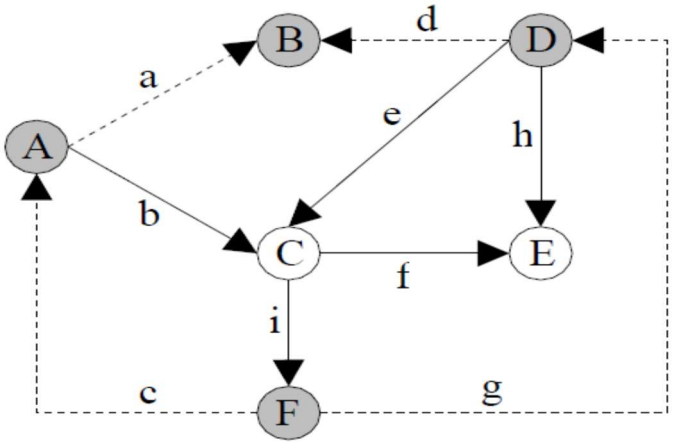
1.1 Cycle et circuit

On appelle ***cycle*** une chaîne non vide telle que le nœud de départ et le nœud d'arrivée sont identiques le (exemple cycle (a; d; g; c) dans la figure 2.1). Si l'on considère un cycle et qu'on lui choisit arbitrairement un sens de parcours, on notera Υ^+ l'ensemble des arcs qui sont dans ce sens de parcours et Υ^- l'ensemble des arcs qui sont dans le sens opposé. Si tous les arcs sont dans le même sens, le cycle est appelé un ***circuit***.

Comme les chaînes et les chemins, un ***cycle élémentaire*** (respectivement un ***circuit élémentaire***) est un cycle (respectivement un circuit) qui ne passe qu'une seule fois par un même nœud. Pour des facilités d'écriture, un cycle élémentaire noté Υ pourra être considéré comme un vecteur d'entiers, ses composantes seront notées Υ_u pour tout arc ***u*** du graphe G, telles que:

$$\gamma_u = \begin{cases} 0, & \text{si } u \notin \gamma \\ +1, & \text{si } u \in \gamma^+ \\ -1, & \text{si } u \in \gamma^- \end{cases}$$

Dans la figure 2.1, le cycle (a; d; g; c) peut être représenté par le vecteur (+1; 0;+1;−1; 0; 0;−1; 0; 0), en supposant les arcs dans l'ordre a, b, c, d, e, f, g, h, i.

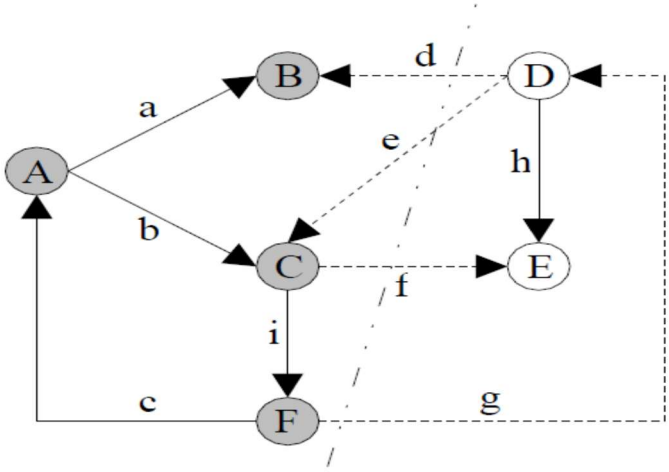


Un exemple de cycle

1.2 Cocycle et cocircuit

Soit A un sous-ensemble des nœuds de G. On note ω(A) le ***cocycle*** de A. Il s'agit de l'ensemble des arcs de G qui ont une extrémité dans A et l'autre dans X \ A, c-à-d les nœuds qui ne sont pas dans A (exemple le cocycle (d; e; f; g) dans la figure 2.2).

L'ensemble ω(A) peut être séparé en deux sous-ensembles: $\omega^+(A)$ qui contient les arcs du cocycle qui ont leur source dans A, et $\omega^-(A)$ qui contient ceux qui ont leur destination dans A. Si tous les arcs sont dans le même sens, le **cocycle** est appelé un ***cocircuit***.



Un exemple de cocycle

De la même manière que le cycle, un cocycle noté **ω** pourra être considéré comme un vecteur d'entiers, ses composantes seront notées ω_u pour tout arc ***u*** du graphe G, telles que:

$$\omega_u = \begin{cases} 0, & \text{si } u \notin \omega \\ +1, & \text{si } u \in \omega^+ \\ -1, & \text{si } u \in \omega^- \end{cases}$$

Dans la figure 2.2, le cocycle (d; e; f; g) peut être représenté par le vecteur (0; 0; 0;−1;−1;+1;+1; 0; 0), en supposant les arcs dans l'ordre a, b, c, d, e, f, g, h, i.

1.3 Lemme des arcs colorés (Minty 1960)

Soit un graphe G = (X;U) avec les arcs sont numérotés de 1 à m et colorés en vert, noir et rouge. Minty a démontré qu'une et une seule des propositions suivantes est vérifiée pour tout arc u= (x; y) ϵ U:

(1) il passe par l'arc 1 un cycle élémentaire uniquement rouge et noir avec tous les arcs noirs orientés dans le même sens ;

(2) il passe par l'arc 1 un cocycle élémentaire uniquement vert et noir avec tous les arcs noirs orientés dans le même sens.

1.4 Base de cycles et base de cocycles

Définitions :

Les cycles C_1 , C_2 ,…,C_k sont des **cycles indépendants** si leurs vecteurs associes vérifient :

$\sum_{i=1..k} \lambda i \, Ci = 0$

Une **base de cycles** est un ensemble de cycles indépendants tel que tout cycle puisse s'écrire comme une combinaison linéaire.

Le **nombre cyclomatique** du graphe est égal à la dimension de la base de cycles C_i .

De même, les cocycles ω_1 , ω_2 ,…, ω_j sont des **cocycles indépendants** si une combinaison linéaire des vecteurs associés ne peut être nulle que si chaque coefficient $\boldsymbol{\lambda}_i$ est nul. Une **base de cocycles** est un ensemble de cocycles indépendants tel que tout autre cocycle puisse s'écrire comme une combinaison linéaire.

Le **nombre cocyclomatique** du graphe est égal à la dimension de la base de cocycles.

Théorème : Soit G un graphe avec **n** sommet, m arcs et p composantes connexes, alors :

$$\nu(G) = m - n + p \quad \text{et} \quad \lambda(G) = n - p$$

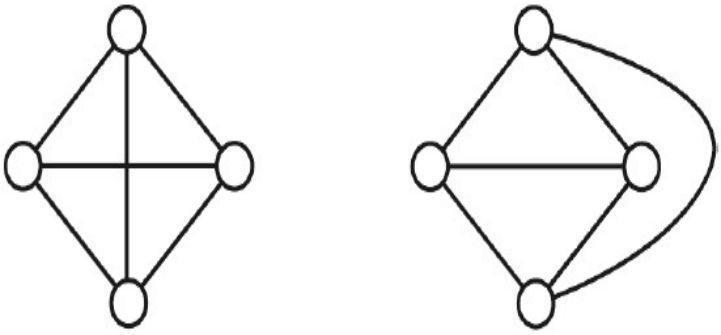
2 Planarité

2.1 Graphe Planaire

Définitions :

Un graphe est **planaire** si on peut le dessiner sur un plan sans que les arêtes (arcs) se croisent.

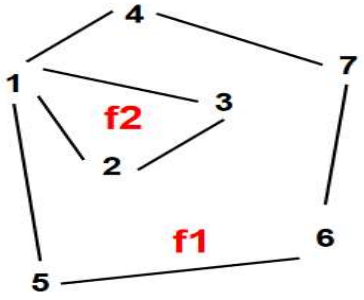
Un **graphe planaire topologique** est une représentation d'un graphe planaire G sur un plan, conformément aux exigences précédentes. Conformément aux exigences précédentes.



Graphe planaire **Graphe planaire topologique**

Dans un graphe planaire topologique, les zones délimitées par des arêtes qui les entourent sont appelées des **faces**.

- Deux faces sont **adjacentes** si elles ont un arc en commun.
- Deux faces sont **opposées** si elles ont un sommet commun sans être adjacentes.
- L'ensemble des arcs qui touchent une face s'appelle la **frontière de la face.**
- On appelle contour d'une face, celui de ces cycles élémentaires qui contient à son intérieur tous les autres arcs de la frontière.
- On notera qu'il y a toujours une face illimitée appelée **face infinie** ou **face externe**.



Contour de f1 = 1-4-7-6-5-1
Frontière de f1=1-3-2-1-5-6-7-4-1

2.2 Formule d'Euler

Théorème : dans un graphe planaire topologique G, les contours des différentes faces finies constituent une base fondamentale de cycles indépendants.

Formule d'Euler : si, dans un graphe planaire connexe de **n** sommets, **m** arêtes et **f** faces, alors: **n - m + f = 2**

Démonstration : le nombre de faces finies est égal au nombre cyclomatique ν (G) d'où :

$$f = \nu(G) + 1 = (m - n + 1) + 1 = m - n + 2.$$

2.3 Théorème de Kuratowski (1930)

Définition :

Une **subdivision** d'un graphe est un graphe obtenu en ajoutant des sommets sur les arcs (ou les arêtes).

Théorème : un multigraphe G est planaire si et seulement s'il ne contient aucune subdivision du graphe complet à 5 sommets K_5 et plus, et du graphe biparti complet $K_{3,3}$

Propriété : les multigraphes complets $K_{3,3}$ et K_5 et plus ne sont pas planaires.

2.4 Graphe Dual

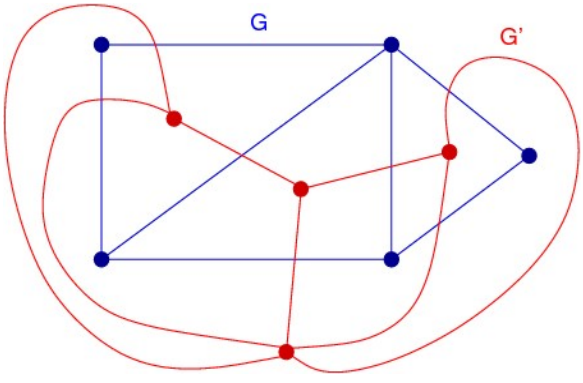
Définition :

Etant donné un graphe planaire G connexe et sans sommets isolés, on peut lui faire correspondre un graphe planaire "**dual**" G^*. Chaque face de G correspond un sommet de G^*. Deux sommets de G^* sont reliés si seulement si les faces correspondantes de G ont une arête commune.

Théorème :

À tout cycle (circuit) élémentaire de G correspond un cocycle (cocircuit) élémentaire de G*, et vice-versa.

Un exemple de graphe dual



3 Arbre, forêt et arborescence

Définition

Un **arbre** est un graphe connexe sans cycle. Une **forêt** est un graphe dont chaque composante connexe est un arbre.

Propriétés

De cette définition découlent les propriétés suivantes.

• Tous les nœuds d'un arbre sont reliés par une chaîne (grâce à la connexité) et une seule (sinon deux chaînes formeraient un cycle).

• Un arbre à n sommets contient exactement n − 1 arcs (car connexe signifie m ≥ n − 1 et sans cycle signifie m <= n − 1).

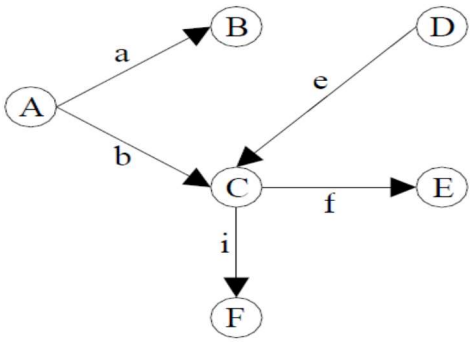
• Un arbre possède au moins deux nœuds de degré 1 (cette propriété se vérifie facilement par récursivité sur la taille de l'arbre).

• L'ajout d'un seul arc dans un arbre introduit un cycle (car m devient supérieur à n − 1).

• La suppression d'un seul arc dans un arbre introduit deux composantes connexes (car m devient inférieur à n − 1).

3.1 Arbre maximal (ou couvrant)

Un **arbre couvrant** d'un graphe G est un sous-graphe de G qui est connexe, sans cycle et contient tous les sommets de G.

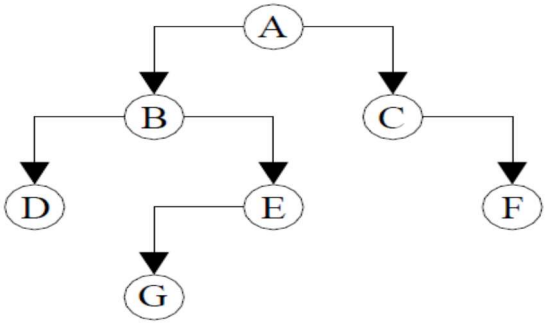


Un exemple d'arbre recouvrant.

3.2 Arbre binaire

Un nœud ***a*** est une ***racine*** du graphe G si pour tout nœud ***x*** de G il existe un chemin de ***a*** à ***x***. Un arbre de racine ***a*** est appelé une ***arborescence*** de racine ***a***.

En particulier, nous empruntons le terme ***arbre binaire*** au domaine des structures de données pour la suite de ce document. Il désigne une arborescence pour laquelle tous les nœuds ont un degré entrant égal à 1 (exceptée la racine pour laquelle c'est 0) et un degré sortant d'au plus 2. Un exemple est montré par la figure 2.5.



Un exemple d'arbre binaire.

Chapitre 3

Flots

1 Flots

Définition

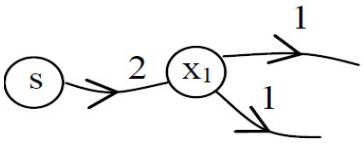
Un flot **f** dans un graphe, associe à chaque arc *u* une quantité **f(u)** qui représente la quantité de flux qui passe par cet arc en provenance de la source vers le puits.

Remarque:

Un flot est conservatif, s'il obéit à la règle de Kirchoff aux sommets suivants: 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.

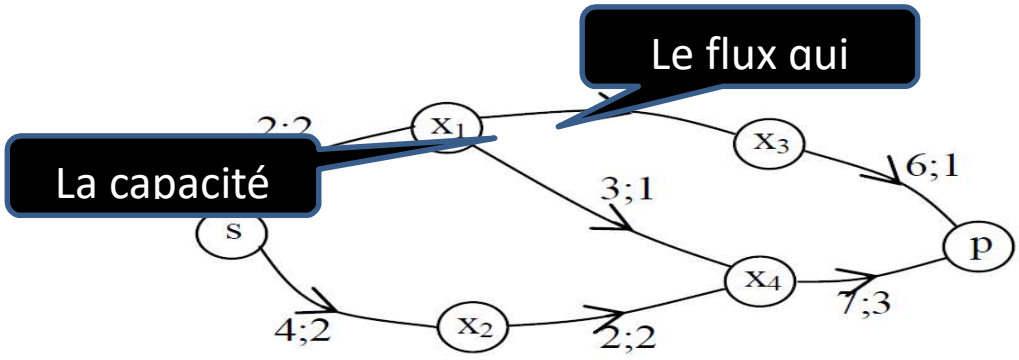


1.1 Flot compatible

Définition

Un flot est **compatible** dans un réseau si pour tout arc **u**=(x,y), 0≤f(u)≤c(u), autrement dit pour chaque arc **u**, le flux qui le traverse ne dépasse pas sa capacité.

Exemple:



1.2 Flot complet

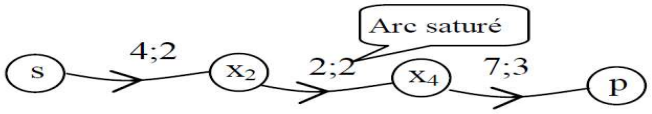
Définition

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

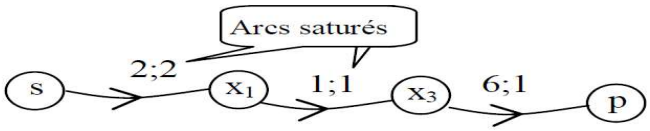
Exemple:

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.

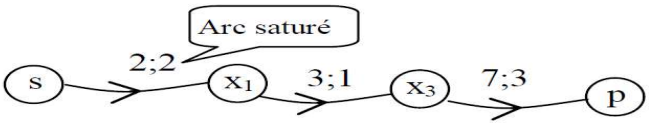
Premier chemin:



Deuxième chemin:



Troisième chemin:



2 Recherche d'un flot maximum dans un réseau de transport

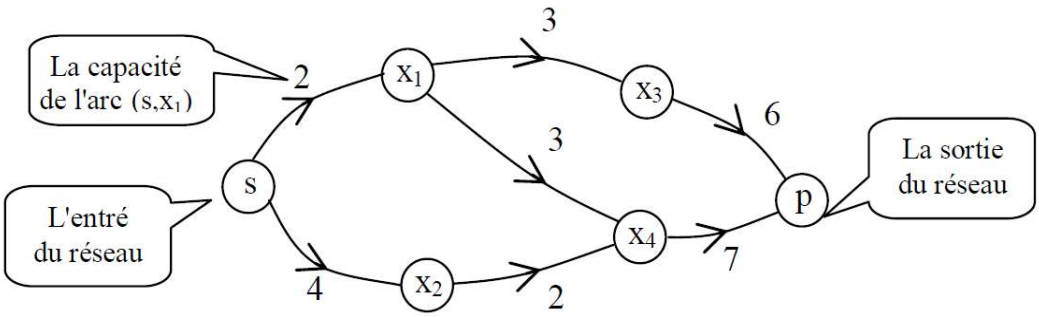
Définition

Un Réseau de Transport est un graphe G= (S,A) orienté, dans lequel chaque arc a une capacité positif c(u) appelé capacité de l'arc u.

On distingue deux sommets particuliers : la source s et le puits p. On supposera, par commodité qu'il existe toujours un chemin permettant d'aller de s à p.

On note un réseau par R=(X ,U,C).

Exemple



- Réseau éléctrique : haute tension, circuit,....
- Réseau routier : transport de marchandises,
- Réseau de fluides, gaz, hydrocarbures,
- Logistique, processus industriels....

Probléme classique: comment maximiser ce flux ?

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

2.2 Théorème de Ford-Fulkerson

Etant donné G=(X,A)un réseau de transport et φ un flot réalisable sur G. La valeur du flot maximum réalisable se**r**vers **p** est égale à la capacité de la coupe minimale séparant S et T.

Max(φ)=Min(c(p))

D'après ce théorème, nous pouvons affirmer ques'il existe un flot φ de valeur v, égale à la capacité d'une coupe (X1,X2) alors le flot est maximum et la coupe est de capacité minimale.

2.3 Algorithme de Ford-Fulkerson

Le principe:

L'idée de l'algorithme 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 chaine pour laquelle le flot peut être augmenté est une chaine 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.

(0) Initialisation

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

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

- Marquer le sommet y successeur de x tel que $(x,y)<c(x,y)$

  On pose: $C^{+}\leftarrow C+\cup \{x,\}$; $A\leftarrow A\cup \{y\}$

- Marquer le sommet y prédécesseur de x tel que $f(x,)>0$

  On pose: $C^{-}:=C^{-}\cup\{x,y\}$; $A\leftarrow A\cup\{y\}$ si on 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 chaine augmentante $C= C^{+}\cup C^{-}$ de s à p.

Pour améliorer le flot on calcule:

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

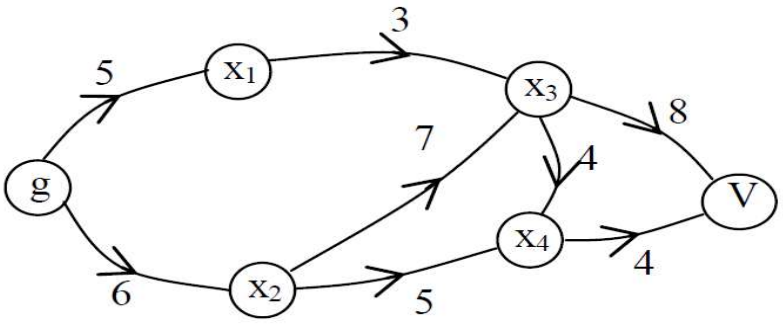
D'où $\varepsilon=min\ [\varepsilon_{1,2}]$ On définit le nouveau flot:

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

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

Exemple

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

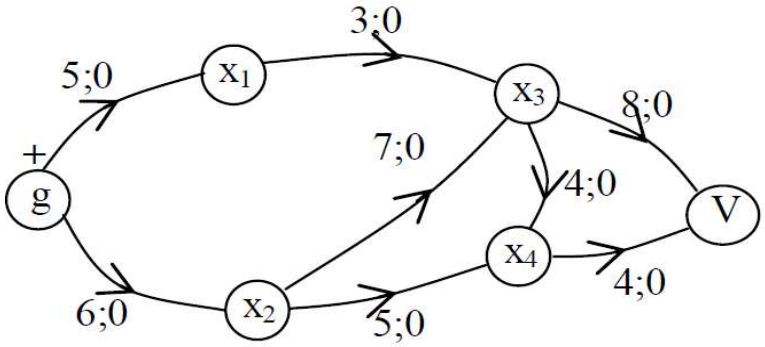


On voudrait connaitre la quantité maximale que peut écouler l'usine. Ce qui revient à chercher un flot maximum sur le réseau.

Initialisation:

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

On pose : $A= \{g\}$; $C^{+} \cup C^{-} = \emptyset \ \ et \ \ f^{k}=0$;

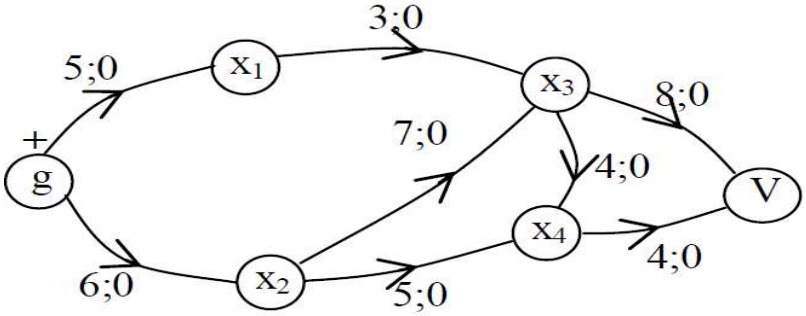


Le réseau R après défini le flot f^{0}

Itération 1:

On marquer tous les sommets de la chaine (g,v) tel que:

- le sommet **y** successeur de **x** et $(x,y) < c(x,y)$
- le sommet **y** prédécesseur de **x** et $f\ (x,) > 0$



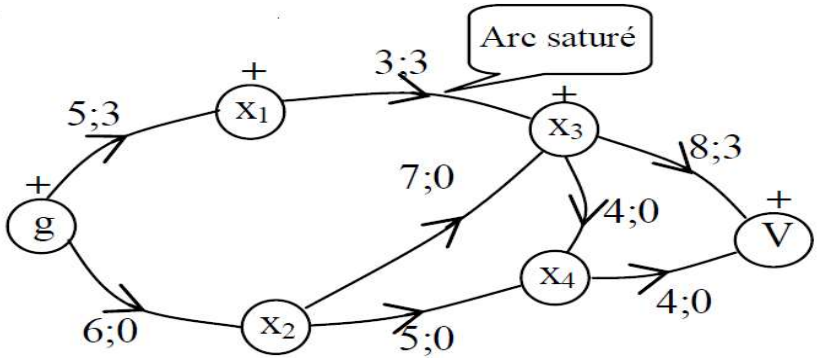
$C = C^+ \cup C^- = C^+ = \{(g,_1)\ ,(x_1,x_3),(x_3,V)\}$, et $A=\{g,x_1,x_3,\mathrm{V}\}$.

On calcule ε_1 :

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

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

$= min\ (5-0;3-0;8-0\ ) = 3$



On améliore ainsi le flot f^0 pour obtenir un nouveau flot f^1, en ajoutant la quantité $\varepsilon 1$ au flot des arcs de C^+.

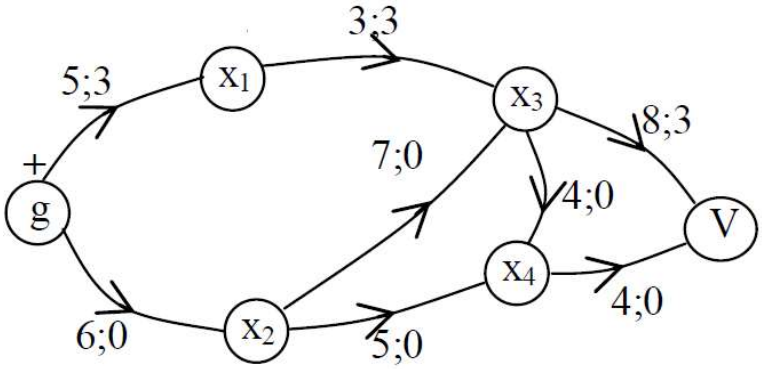
Itération 2:

En reprend l'instruction

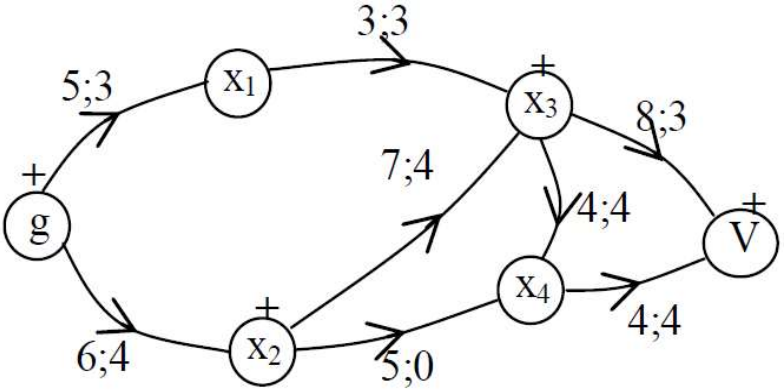
$C=C^+\cup C^-=C^+=\{(g,x_2),(x_2,x_3),\ (x_3,\ x_4),(x_4,V)\}, A=\{g,x_2,x_3,\ x_4\ \mathrm{V}\}$

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

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



On améliore ainsi le flot f^1 pour obtenir un nouveau flot f^2, en ajoutant la quantité $\varepsilon 1$ au flot des arcs de C^+.



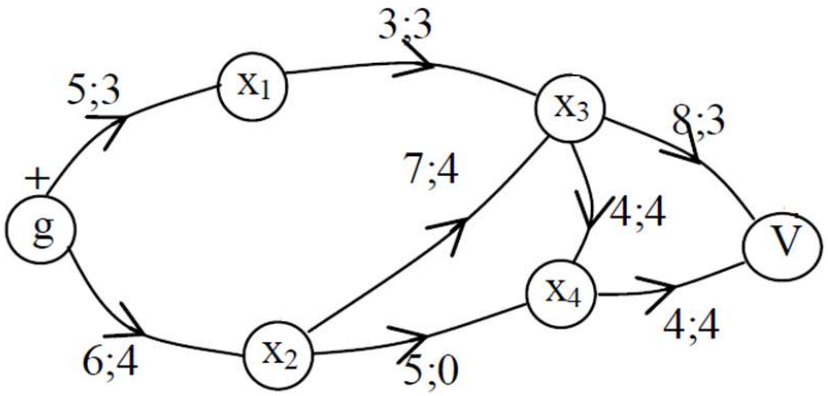
Itération 3:

En reprend l'instruction

$C=C^{+}\cup C^{-}=\ \{(g,x_2),(x_2,x_4),(x_3,V)\}\cup\{(x_4,x_3)\},\quad A=\{g,x_2,x_3,x_4,\mathrm{V}\}$

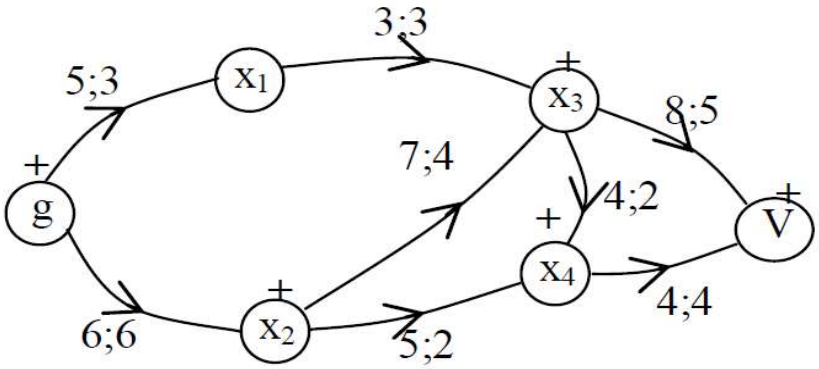
Calcule de $\varepsilon_1=min\ [(u)-f(u);\ u\in C^{+}]=min[6-4;5-0;8-3]=2$

$\varepsilon_2=min\ [(u)\ ;\ u\in C^{-}]=min[f^{2}\ (x_3,x_4)]=4$



$\varepsilon=min\ [\varepsilon_{1,2}]=[2,4]=2$

On améliore ainsi le flot f^{2} pour obtenir un nouveau flot f^{3}, en ajoutant la quantité $\varepsilon 1$ au flot des arcs de C^{+}, et retranchant la quantité $\varepsilon 1$ au flot des arcs C^{-}.



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	$(g,1)$	$(g,2)$	$(x_1,3)$	$(x_2,3)$	$(x_2,4)$	$(x_3,4)$	(x_2,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-à-d: $f_{max}=\sum((s,x)/x\in\Gamma^{+}(s))=\sum f(x,p)/x\in\Gamma^{-}(p))$

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

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

Chapitre 4

Problèmes de cheminement

1 Recherche des composantes connexes

1.1 Présentation des objectifs

La vérification de la connexité d'un graphe est un des premiers problèmes de la théorie des graphes. De nombreux algorithmes permettent de trouver l'ensemble des sommets appartenant à une même composante connexe.

(Trémeaux 1882-Tarjan 1972).

1.2 La recherche des composantes connexes

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

Le principe

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, et la liste de ses composantes connexes.$\{C_1, C_2,..C_k\}$

(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 de sommets marqués.

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

(1.4) On teste si W=Ø. - Si oui terminer aller à (2).

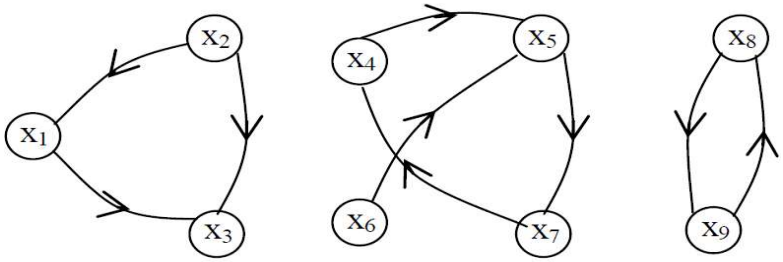
- Si non aller à (1)

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

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

Application:

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

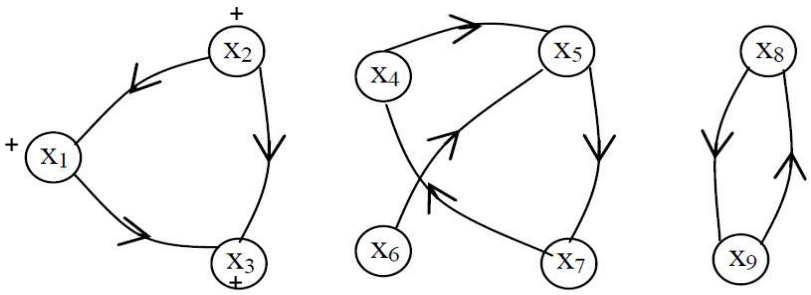


Le graphe (G) n'est pas connexe, car il n'existe pas de chaîne reliant les sommets x_3 et x_8.

Initialisation: k=0, W={x_1,x_2,x_3,…x_9}

Itération 1:

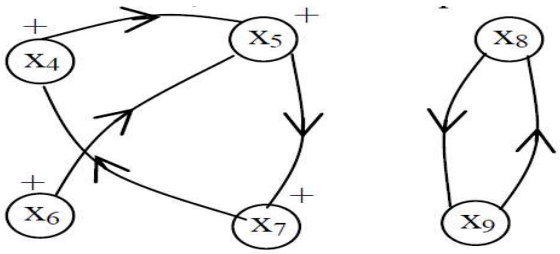
On choisir dans W le sommet x_2, et on le marque d'un (+), on marque ensuite ses voisins x_1 et x_3.



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} ≠ ϕ

Itération 2:

on choisir dans W le sommet x_5, et on le marque d'un (+), on marque ensuite ses voisins x_4 ,x_6 et x_7



Soit C2={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} ≠ ϕ

Itération 3:

On choisir dans W le sommet x_8, et on le marque d'un (+), on marque ensuite ses voisins x_9

2 Recherche du plus court chemin

2.1 La recherche de plus court chemin

Si un plus court chemin 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 **π(x).**

Présentation des conditions

Soit un graphe G, on associe à chaque arc ***u*** de G une **longueur** l(u) telle que :

$\forall$ ***u*** $\in$ U , l(u) $\geq$ 0.

Soit ***a*** et ***b*** deux sommets de G, il s'agit de trouver un chemin **μ**(*a,b*) telle que:

$l(\mu)=\sum_{u\in\mu} l(u)$, μ(a,b)est appelé le **plus court chemin** de ***a*** à **b**.

2.2 Algorithme de Dijkstra

Principe de l'algorithme :

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 .

L'idée 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.

Algorithme 7: Algorithme de Dijkstra

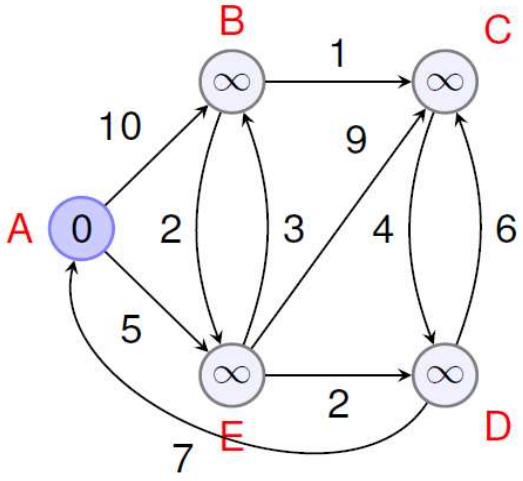
Données : Un graphe orienté pondéré $G = (X, A, W)$ et un sommet $s \in X$
Résultat : Le plus court chemin de s vers tous les autres sommets de G

```
// V : Tableau stockant les étiquettes des sommets de G
Initialiser V à +∞
V[s] = 0
// P : Tableau permettant de retrouver la composition des chemins
Initialiser P à 0
P[s] = s
répéter
    // Recherche du sommet x non fixé de plus petite étiquette
    V_min = +∞
    pour y allant de 1 à N faire
        si y non marqué et V[y] < V_min alors
            x ← y
            V_min ← V[y]
    // Mise à jour des successeurs non fixés de x
    si V_min < +∞ alors
        Marquer x
        pour tout successeur y de x faire
            si y non marqué et V[x] + W[x,y] < V[y] alors
                V[y] = V[x] + W[x,y]
                P[y] = x
jusqu'à V_min = +∞
```

Exemple:

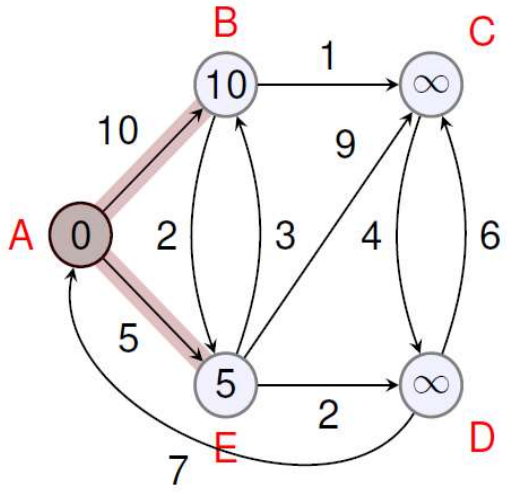
Cherchons les plus courts chemins d'origine A dans ce graphe:

On se place au sommet de plus petit poids, ici le sommet A.



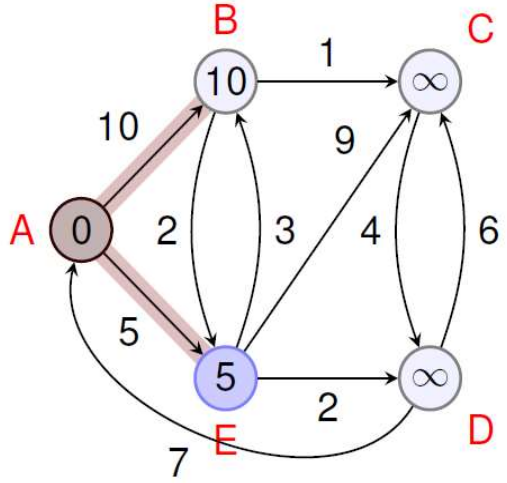
A	B	C	D	E
0	∞	∞	∞	∞
•				
•				
•				
•				
•				

On étudie chacune des arêtes partant du sommet choisi. Dans les colonnes, on met la distance à A, et le sommet d'où l'on vient.



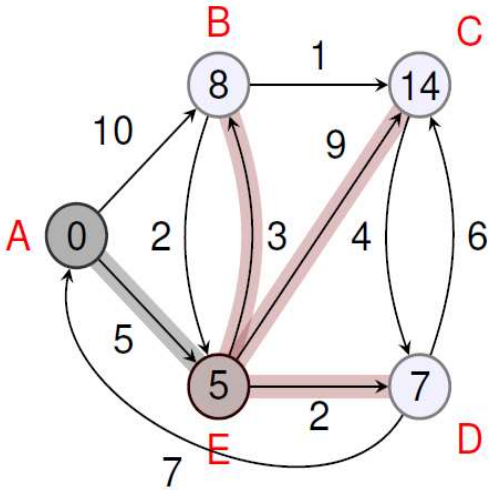
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•				
•				
•				
•				

On se place de nouveau au sommet de plus petit poids, ici E.



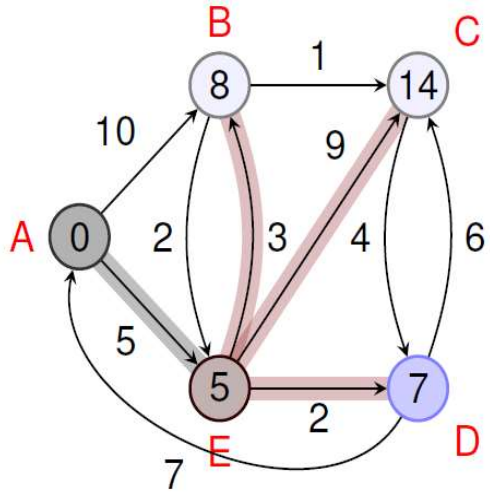
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•				•
•				•
•				•
•				•

En applique l'algorithme.



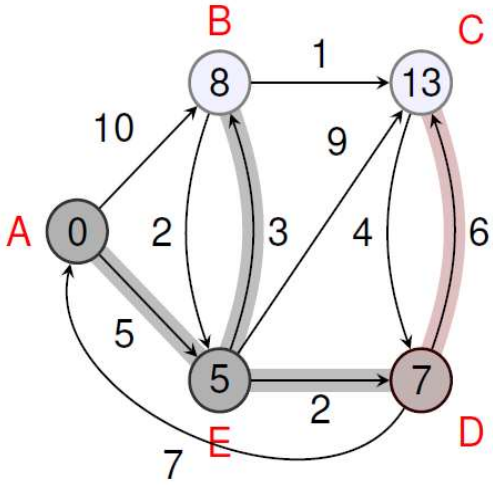
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•				•
•				•

On se place de nouveau au sommet de plus petit poids, ici D.



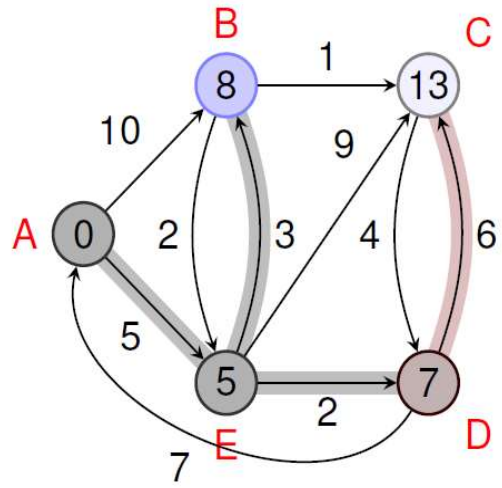
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	$\boxed{7_E}$	•
•			•	•
•			•	•

En applique l'algorithme



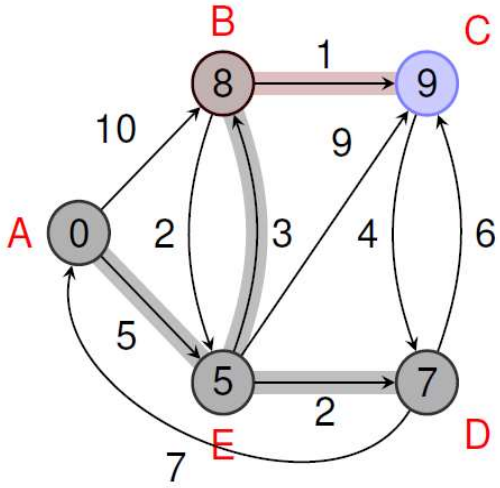
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	$\boxed{7_E}$	•
•	8_E	13_D	•	•
•			•	•
•			•	•

On se place de nouveau au sommet de plus petit poids, ici B.



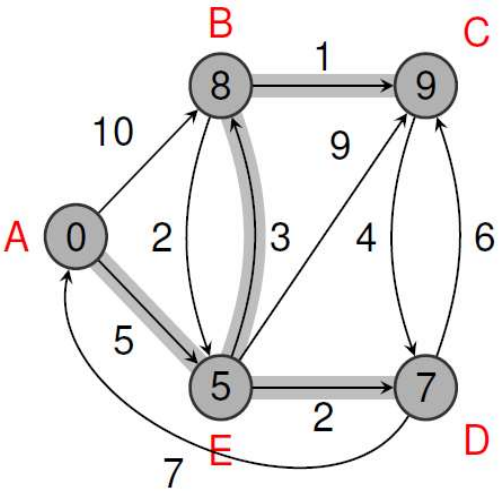
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	$\boxed{8_E}$	13_D	•	•
•	•		•	•
•	•		•	•

En applique l'algorithme.



A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	8_E	13_D	•	•
•	•	$\boxed{9_B}$	•	•
•	•	•	•	•

Si l'on ne considère que les flèches soulignées, on obtient un arbre, un graphe sans cycle.



A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	8_E	13_D	•	•
•	•	9_B	•	•
•	•	•	•	•

3 Recherche d'un arbre de poids extrémum

3.1 Présentation des objectifs

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

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.

3.2 Algorithme de Kruskal 1956

Données :

– Graphe $G = (V,E)$ ($|V| = n$, $|E| = m$)

– Pour chaque arête e de E , son poids $c(e)$.

Résultat :

Arbre ou forêt maximale $A = (V,F)$ de poids minimum.

Trier et renuméroter les arêtes de G dans l'ordre croissant de leur poids :

$c(e_1) \leq c(e_2) \leq \ldots \leq c(e_m)$.

Poser $F \leftarrow \emptyset$, $k \leftarrow 0$;

Tant que $k < m$ **et** $|F| < n - 1$ **faire**

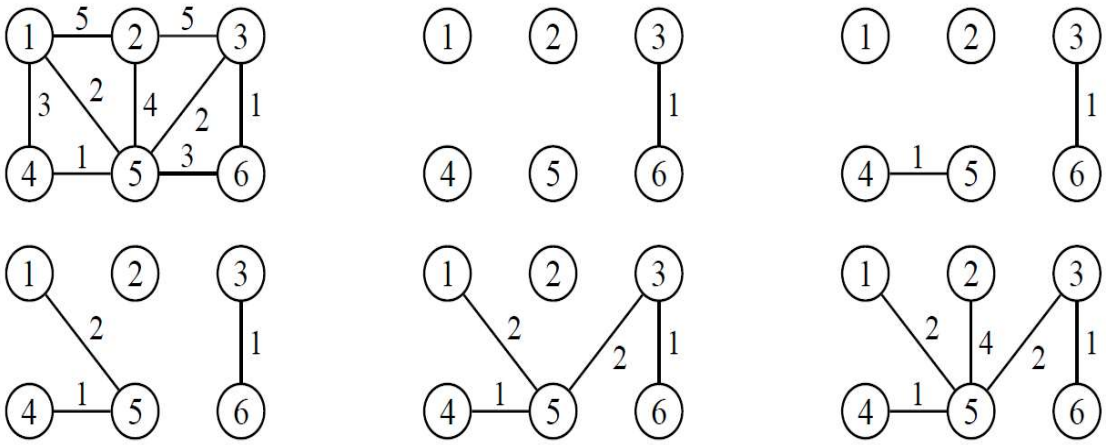
Début

si e_{k+1} ne forme pas de cycle avec F **alors**

$F \leftarrow F \cup \{e_{k+1}\}$; $k \leftarrow k+1$;

Fin

Exemple:



Les arêtes de poids 3 n'ont pas pu être placées, car elles auraient formé un cycle. L'algorithme s'est arrêté dès que cinq arêtes ont été placées. Toute arête supplémentaire aurait créé un cycle.

S'il y a plusieurs arêtes de même poids, il peut y avoir plusieurs arbres couvrants de poids minimum: tout dépend de l'ordre dans lequel ces arêtes ont été triées.

Chapitre 5

Problèmes Hamiltonien et Eulérien

1 Problème Hamiltonien

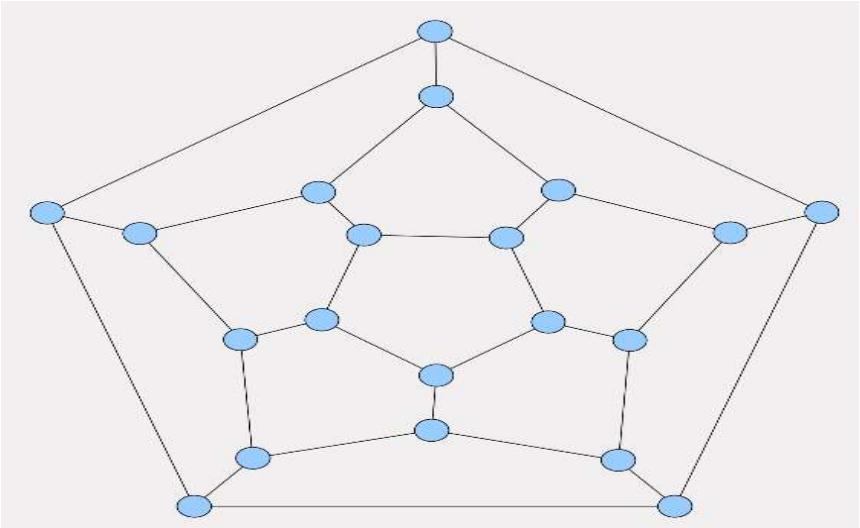
1.1 Définitions

Soit G=(V,E)un graphe orienté ou non. Une chaîne, un chemin, un cycle ou un circuit de G sont dits hamiltoniens s'ils passent une et une seule fois par tous les sommets de G.

Un graphe hamiltonien est un graphe non orienté possédant un cycle hamiltonien ou un graphe orienté possédant un circuit hamiltonien.

Cette appellation vient d'un jeu proposé par Sir W. Hamilton en 1859 : un voyageur veut visiter 20 villes aux sommets d'un dodécaèdre en passant une fois et une seule par chaque ville et en revenant à son point de départ.

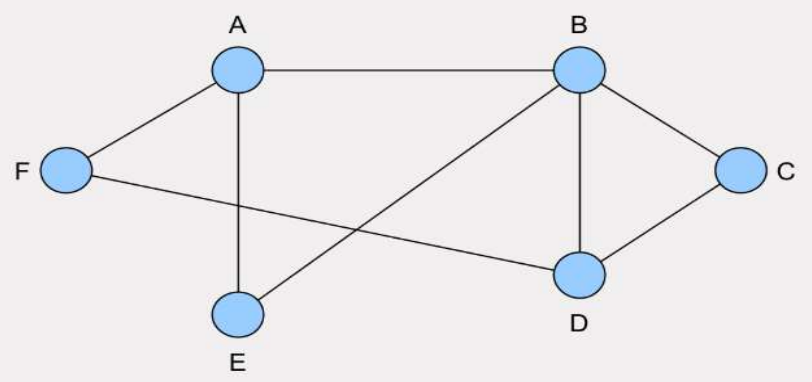
La question est la suivante : peut-on effectuer un parcours en passant une et une seule fois par chacune de ces villes et en revenant à son point de départ ?



On peut associer à ce solide un **graphe planaire**, dont les sommets et les arêtes correspondent à ceux du dodécaèdre.

Exemple

Considérons le graphe non orienté dont la représentation sagittale est la suivante :



Ce graphe est hamiltonien car il possède le cycle (A,E,B,C,D,F,A).

1.2 Condition nécessaire d'existence d'un cycle hamiltonien

Si l'on ne connait pas encore de conditions nécessaires pour affirmer qu'un graphe est hamiltonien ou non, il existe quelques **conditions suffisantes** non **restrictives**.

Condition 1 :

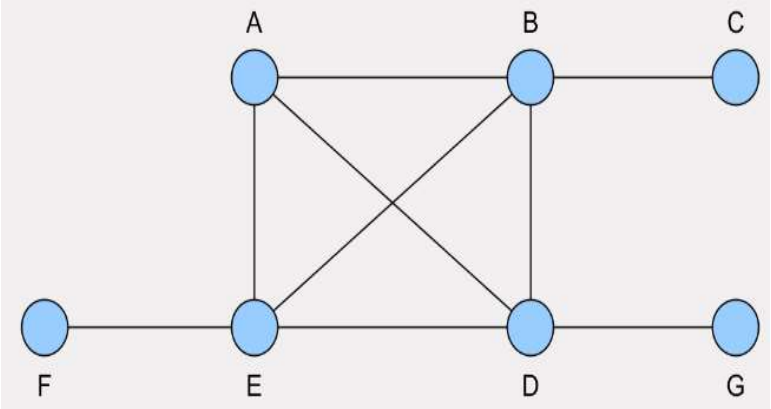
Soit G=(V,E) un graphe **non orienté**.

Si G possède un sommet de degré 1 il ne peut pas être hamiltonien.

Ce résultat est assez évident, car pour qu'un graphe possède un cycle hamiltonien on doit pouvoir arriver et repartir par n'importe quel sommet, et donc le degré de chacun d'eux doit être au moins égal à 2.

Exemple d'un graphe non hamiltonien

Considérons le graphe non orienté dont la représentation sagittale est la suivante :

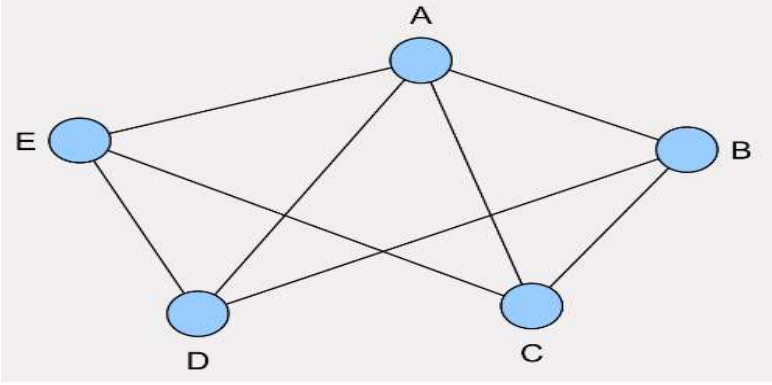


Ce graphe n'est pas hamiltonien car le F est de degré 1.

Condition 2 : Condition de Dirac Gabriel Andrew Dirac (1925-1984)

Soit G=(V,E) un graphe **non orienté** d'ordre n, avec n≥3. Si pour tout sommet de G on a d(x)≥ n/2 G est hamiltonien.

Exemple :

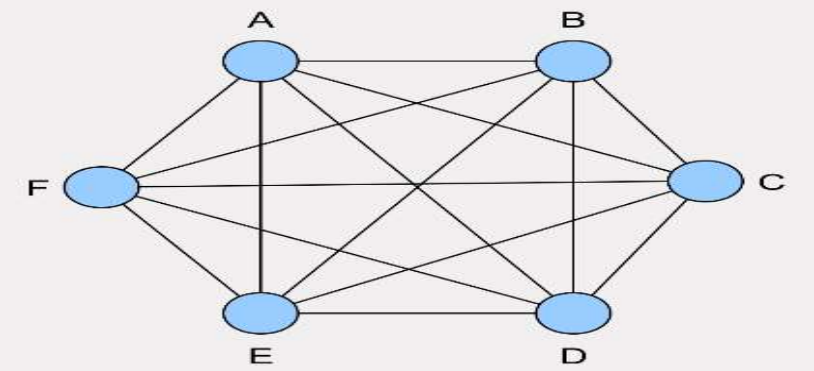


Ce graphe est d'ordre 5 et chacun de ses sommets a un degré au moins égal à 3. Il est donc hamiltonien.

Condition 3 :

Soit G=(V,E) un graphe **non orienté** d'ordre n, avec n≥3. Si le graphe G est complet alors il est hamiltonien.

Exemple



Démonstration :

Le degré de chaque sommet est égal à n−1 donc la condition de Dirac est vérifiée.

2 Problème Eulérien

Le problème eulérien est un des plus vieux problèmes combinatoires comme le prouve la résolution par Euler du problème des 7 ponts de Königsberg.

2.1 Définitions

Un graphe G est **Eulerien** si et seulement si :

- dans le cas où G est un graphe non-orienté chaque sommet x de G doit avoir un degré d(x) pair.
- dans le cas où G est un graphe orienté chaque sommet x de G doit avoir des degrés entrants et sortants égaux $d^+(x)=d^-(x)$

Un multigraphe est un **graphe eulérien** s'il contient un cycle eulérien.

2.2 Condition nécessaire et suffisante d'existence d'une chaîne eulérienne

Théorème d'Euler (1766) :

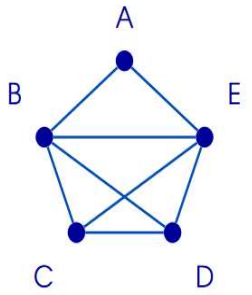
Un graphe connexe G possède une chaîne eulérienne ssi le nombre de sommets de degré impair est 0 ou 2.

Un graphe connexe G possède un cycle Eulérien si et seulement si chaque sommet est de degré pair.

On appelle **chaîne eulérienne** (**cycle eulérien**), une chaîne (un cycle) utilisant une fois et une seule chacune des arêtes de G.

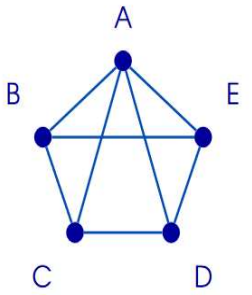
Un multigraphe est un **graphe eulérien** s'il contient un cycle eulérien.

Exemple:



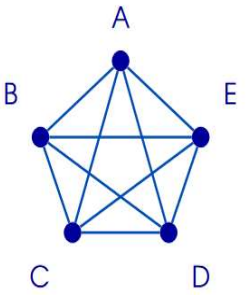
Graphe G_1

G_1 a une chaine eulérienne : C-B-A-E-B-D-E-C-D.
G_1 n'a pas de cycle eulérien.
G_1 **n'est pas donc eulérien.**



Graphe G_2

G_2 n'a pas de chaîne eulérienne.
G_2 n'a pas de cycle eulérien.
G_2 **n'est pas donc eulérien.**



Graphe G_3

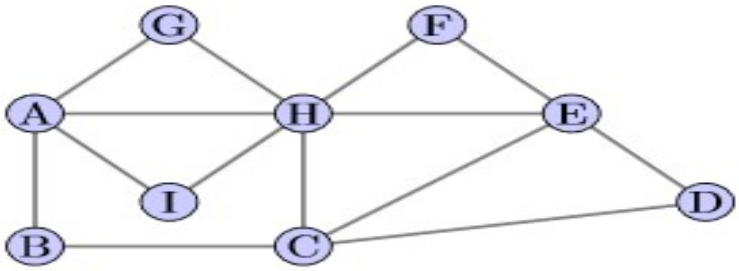
G_3 a plusieurs chaînes eulériennes et plusieurs cycles eulériens. Par exemple : A-C-E-A-B-D-E-B-C-D-A.
G_3 **est donc eulérien.**

Algorithme d'Euler :

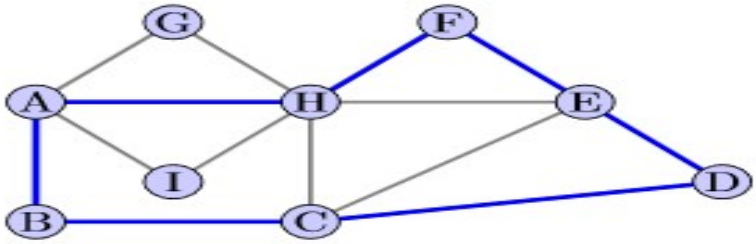
Entrées : Un graphe ayant 0 ou 2 sommets de degrés impairs
début
  si *il y a deux sommets de degré impair* **alors**
    on construit une chaîne reliant ces deux sommets;
  sinon
    on construit un cycle quelconque à partir d'un sommet;
  tant que *il reste des arêtes non parcourues* **faire**
    on choisit un sommet de la chaîne précédente et on construit un cycle fermé à partir de ce sommet, cycle n'empruntant que des arêtes non parcourues.;
fin

Exemple :

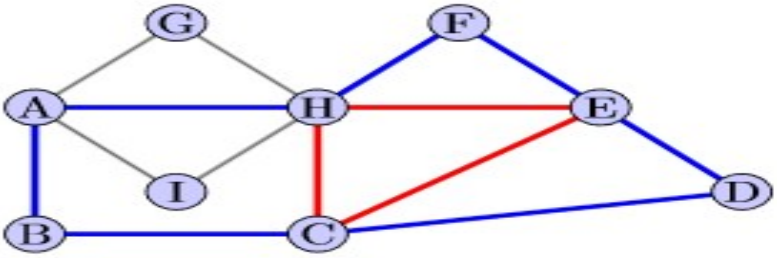
On remarque que tous les sommets du graphe sont de degrés pairs ; il existe donc un cycle eulérien. Nous allons appliquer l'algorithme d'Euler pour en déterminer un.



On commence par tracer un cycle à partir du sommet A : ABCDEFHA. On obtient alors le graphe ci-dessous :



Puis, à partir du sommet E on ajoute le cycle EHCE pour obtenir le graphe ci-dessous :



Puis, à partir du sommet A on ajoute le cycle AIHGA pour obtenir le graphe ci-dessous :

Finalement le cycle eulérien est ABCDEHCEFHAIHGA

2.3 Lien entre problème eulérien et hamiltonien

Proposition :

Un graphe eulérien sans subdivision est hamiltonien mais la réciproque est évidemment fausse.

Chapitre 6

Coloration

1 Coloration

1.1 Introduction

Les problèmes de coloration de graphes apparaissent dans de nombreux contextes. Un premier exemple est celui de la planification d'une session d'examens d'une université. Si V est l'ensemble des cours et si une arête lie deux cours ayant au moins un étudiant en commun, le nombre chromatique est le nombre minimum de périodes permettant de planifier des examens, de telle sorte que chaque étudiant puisse passer tous ses examens. Un deuxième exemple bien connu est celui de la coloration des pays d'une carte de géographie, de telle sorte que deux pays voisins, c'est-à-dire ayant une frontière commune, soient de couleurs différentes.

1.2 Coloration des sommets

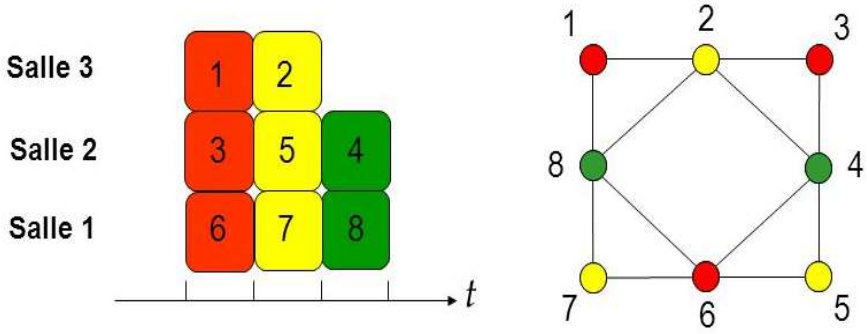
Une « coloration des sommets » d'un graphe G est une affectation de couleurs aux sommets telle que chaque arête de G ait ses deux extrémités de couleur différente. On cherche généralement à déterminer une coloration utilisant aussi peu de couleurs que possible.

Le plus petit nombre de couleurs nécessaire pour colorer les sommets d'un graphe G est appelé « **le nombre chromatique** » de G et est noté **χ(G)**.

Un graphe est dit **p-chromatique** si ses sommets admettent une coloration en **p** couleurs.

Exemple :

Obtenir un bon emploi de temps c'est un problème très difficile



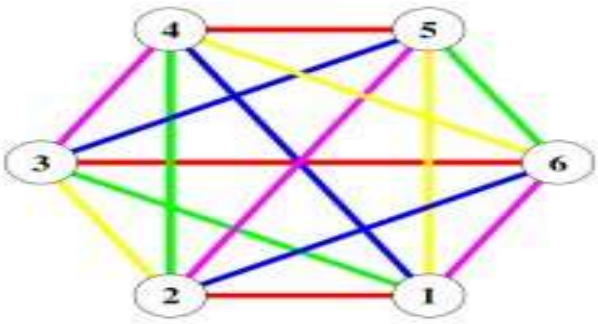
1.3 Coloration des des arêtes

Une « coloration des arêtes » d'un graphe G est une affectation de couleurs aux arêtes telle que les arêtes ayant une extrémité en commun soient de couleur différente. On cherche généralement à déterminer une coloration utilisant aussi peu de couleurs que possible.

Le plus petit nombre de couleurs nécessaires pour colorer les arêtes d'un graphe G s'appelle « **l'indice chromatique** » de G et est noté **q(G)**.

Problème:

Dans une petite classe de six étudiants, certains élèves sont amis et d'autres non. La relation d'amitié est réciproque : si *A* est ami avec *B*, alors *B* est ami avec *A*.



On peut écrire ce problème en termes de graphes, en prenant 6 sommets pour les six étudiants et en reliant deux étudiants dès qu'ils sont amis.

1.4 Propositions

Propriété 1: Le nb chromatique du graphe complet K_n est **n**.

Propriété 2: Soit G un graphe. Si G contient un sous-graphe complet d'ordre n, alors **$\gamma(G) > n$**

Propriété 3: Soit G un graphe d'ordre n, alors **$\gamma(G) \leq n$**.

Propriété 4: Soit G un graphe et soit $\omega(G)$ l'ordre de sa plus grande clique (Une clique est un graphe dans lequel chaque paire de sommets est liée par une arête), alors **$\gamma(G) \geq \omega(G)$.**

Théorème de Brooks : Considérons un graphe G et soit r le plus grand des degrés des sommets. Alors : **$\gamma(G) \leq r+1$**.

Théorème de Berge (1970): l'indice chromatique d'un graphe biparti de degré maximum r est ***q*(G)= r**.

2 Algorithmes de coloration

2.1 Algorithme glouton

Les algorithmes gloutons sont couramment utilisés dans la résolution de problèmes. Le principe de tels algorithmes consiste à choisir des solutions locales optimales d'un problème dans le but d'obtenir une solution optimale globale au problème. Dans le cas du coloriage de graphe, cela va consister à colorier les sommets un par un avec la plus petite couleur possible ; l'ensemble des couleurs possibles étant donné par les couleurs de ses voisins. L'ordre dans lequel les sommets sont traités définit les différentes variantes de d'algorithme. Une méthode est de colorer les sommets par ordre de difficultés décroissantes.

2.2 Algorithme glouton de Welsh & Powell

2.2.1 Principe

L'algorithme de Welsh & Powell consiste ainsi à colorer séquentiellement le graphe en visitant les sommets par ordre de degré décroissant.

L'idée est que les sommets ayant beaucoup de voisins seront plus difficiles à colorer, et donc il faut les colorer en premier.

Il y a une certaine procédure à respecter afin de mettre en œuvre cet algorithme.

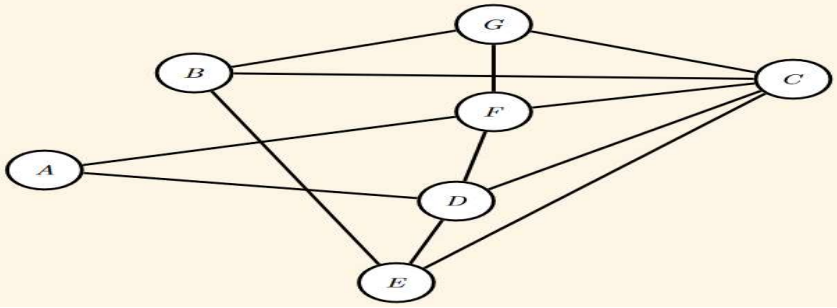
2.2.2 Procédure

1) Repérer le degré de chaque sommet.
2) Ranger les sommets par ordre de degrés décroissants
3) Attribuer au 1er sommet (A) de la liste une couleur.
4) Suivre la liste en attribuant la même couleur au premier sommet (B) qui ne soit pas adjacent à (A).
5) Suivre (si possible) la liste jusqu'au prochain sommet (C) qui ne soit adjacent ni à A ni à B.
6) Continuer jusqu'à ce que la liste soit finie.
7) Prendre une deuxième couleur pour le premier sommet (D) non encore coloré de la liste.
8) Répéter 4 à 7 et continuer jusqu'à coloré tous les sommets

Exemple

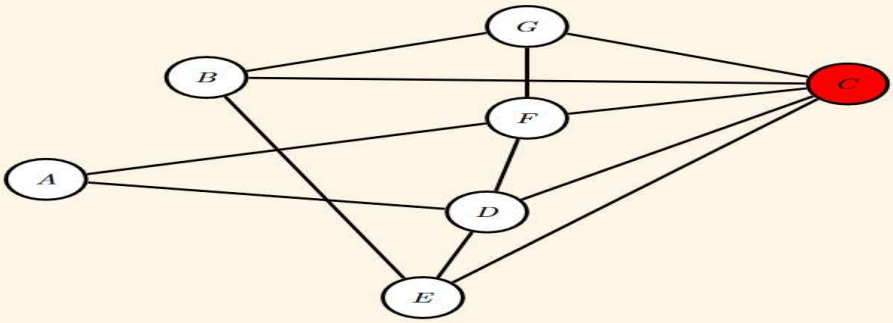
On peut classer les sommets par degré décroissant, puis les colorer :

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



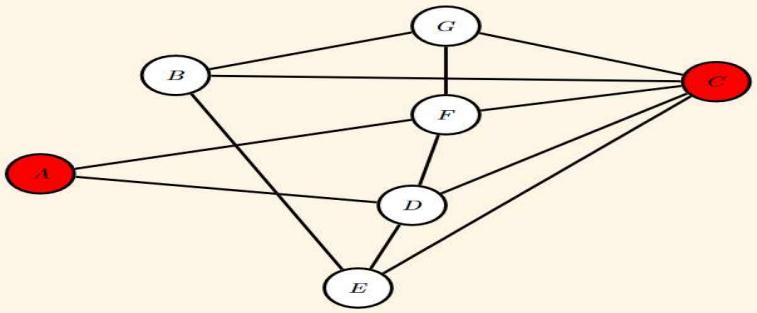
Une coloration pourrait être :
on colorie C en rouge; puis dans l'ordre décroissant...

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



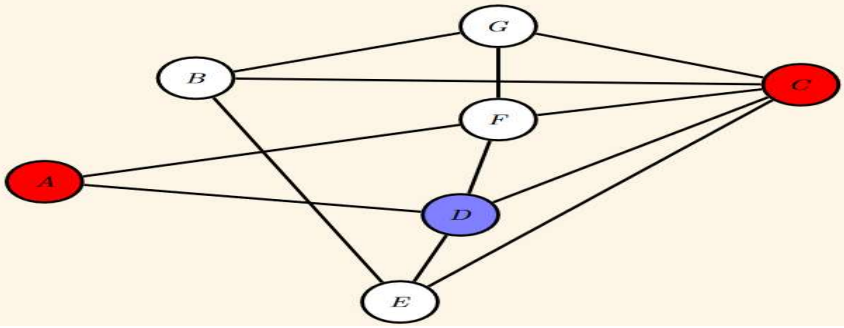
Une coloration pourrait être :
on colorie C en rouge; puis dans l'ordre décroissant, le sommet non adjacent A en rouge.

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



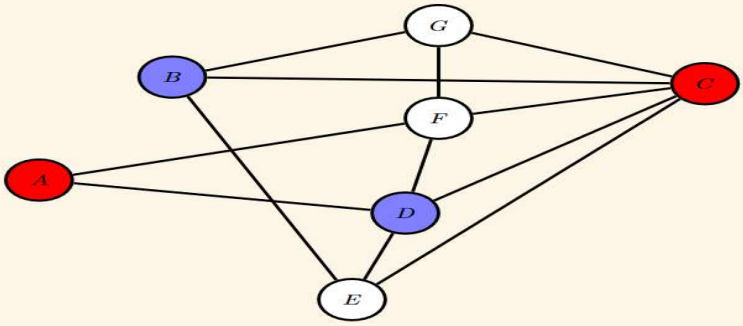
Puis on change de couleur, on colorie D en bleu; puis dans l'ordre décroissant...

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



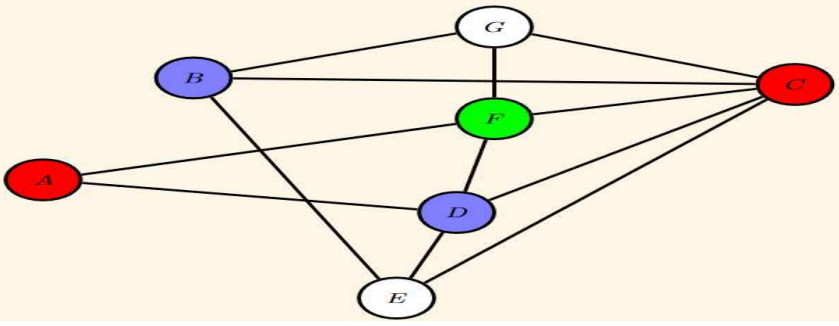
Puis on change de couleur, on colorie D en bleu; puis dans l'ordre décroissant, le sommet non adjacent B en bleu.

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



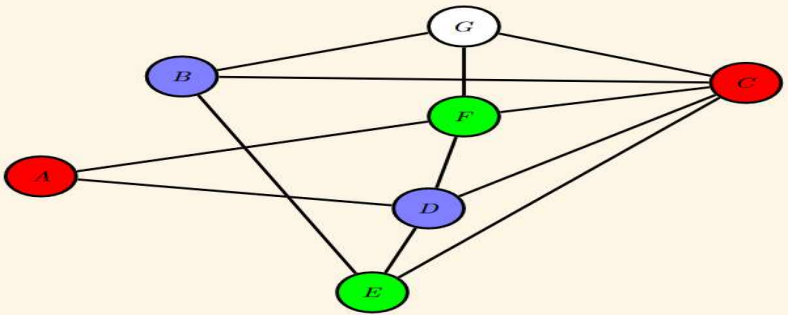
Puis on change de couleur, on colorie F en vert; puis dans l'ordre décroissant...

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



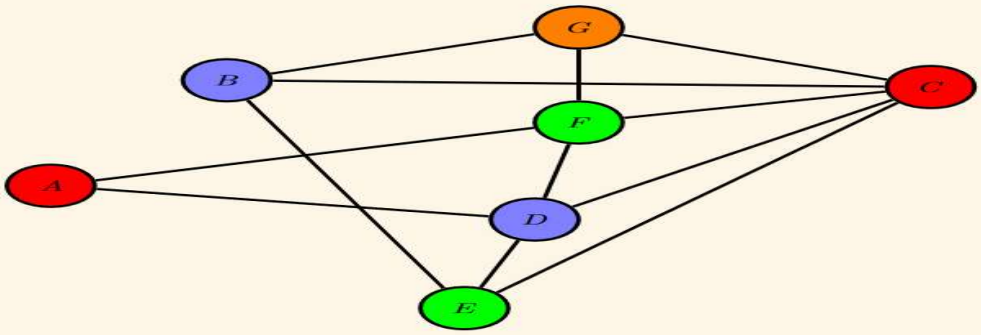
Puis on change de couleur, on colorie F en vert; puis dans l'ordre décroissant, le sommet non adjacent E en vert.

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



Enfin, on change de couleur, on colorie G en orange.

Point	C	D	F	B	E	G	A
Degré	5	4	4	3	3	3	2



2.3 Algorithme DSATUR

2.3.1 Principe

DSATUR est un algorithme de coloration de graphes créé par Daniel Brélaz en 1979 à l'EPFL (École polytechnique fédérale de Lausanne).

Il s'agit d'un algorithme de coloration séquentiel par heuristique.

DSAT est l'abréviation de degré saturation et calculer de la manière suivante :

- Si aucun voisin de x est colorié, alors DSAT(x) = le degré de x
- Sinon, DSAT(x) = le nombre de couleurs différentes utilisées dans le voisinage de x

2.3.2 Procédure

Au départ le graphe n'est pas colorié

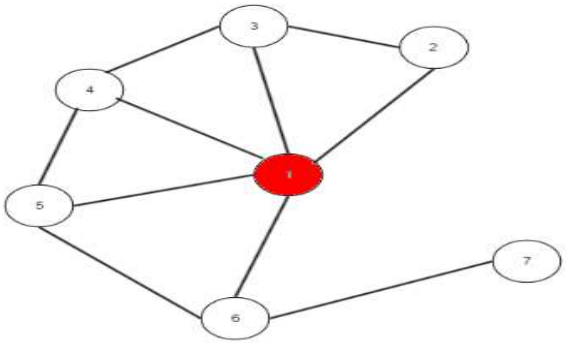
1) Ordonner les sommets par ordre décroissant de degré.
2) Colorer un sommet de degré maximum avec la couleur 1.
3) Choisir un sommet non coloré avec DSAT maximum. Si conflit choisir celui avec degré maximum.
4) Colorer ce sommet par la plus petite couleur possible
5) Si tous les sommets sont colorés alors stop. Sinon aller en 3.

2.3.3 Exemple

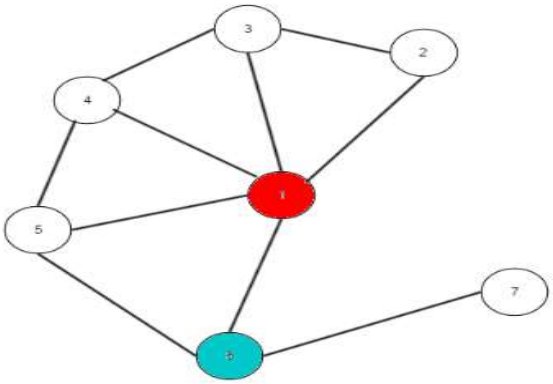
Soit les couleurs hiérarchisées :

On prend le sommet de plus haut degré,

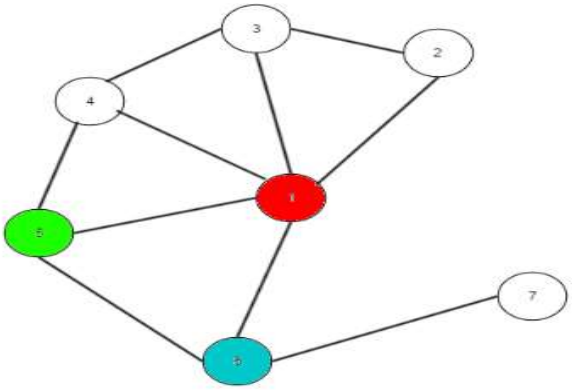
soit le sommet 1. On lui attribue la couleur minimale, soit C1 :



Puis, on regarde plus un voisin du sommet 1 étant de plus haut degré: Nous allons prendre le sommet 6 et en lui attribuer une couleur minimale, mais différente de la couleur du sommet : On prend C2 :



On fait la même chose pour le voisin du sommet étant de plus haut degré : le sommet 5. On lui attribue une couleur minimale différente de celle de ses voisins : la couleur C3 :



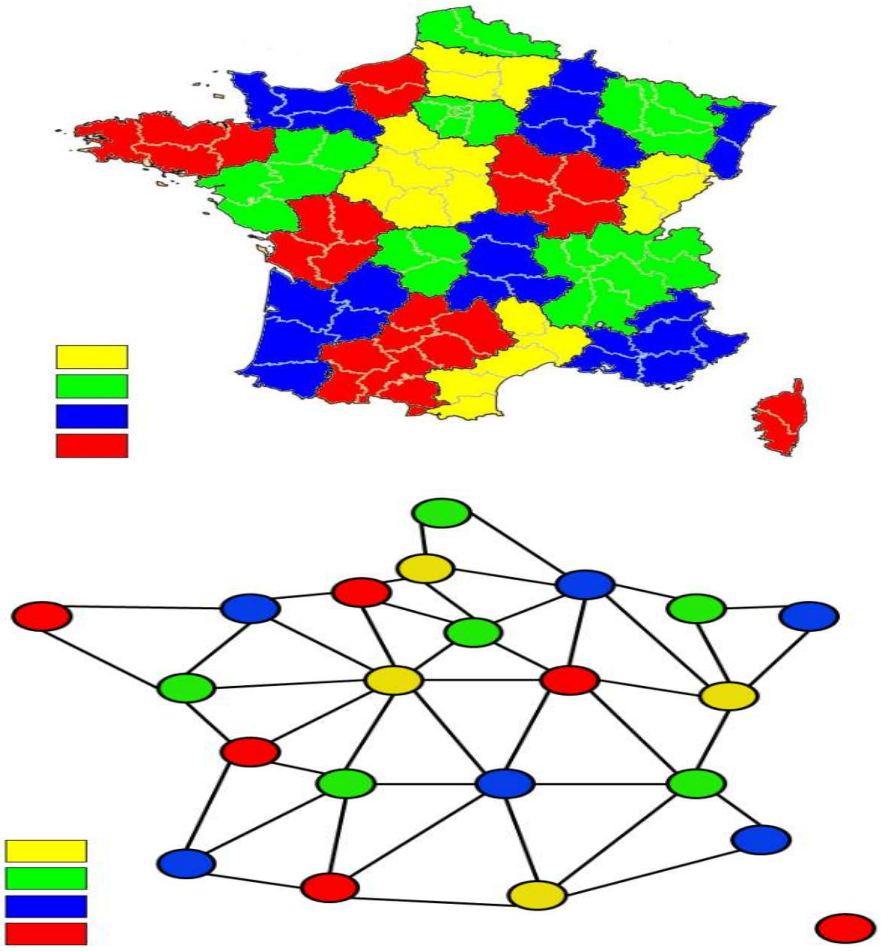
3 Le théorème des 4 couleurs

3.1 Peu d'histoire

L'un des problèmes les plus célèbres dans la théorie des graphes est le problème des 4 couleurs. Ce problème est resté pendant plus d'un siècle sans solution. En 1852, F. Guthrie s'est demandé s'il est possible de colorer une carte de géographie à l'aide de quatre couleurs de sorte à ce que des pays voisins aient des couleurs différentes?

De nombreux travaux ont été publiés lors du siècle suivant pour réduire le nombre de couleurs à quatre, jusqu'à la démonstration finale de deux chercheurs américains, K. Appel et W. Haken en 1976. Ils ont pu répondre affirmativement à cette conjecture des quatre couleurs.

Exemple



3.2 Annonce

Théorème :

Tout graphe planaire est 4 chromatique

Application :

On peut colorier toute carte de géographie avec 4 couleurs, de sorte que deux régions ayant une frontière commune soient de différentes.

Bibliographie

[1] C.P. Bruter, Les matroides, nouvel outil mathématique, Initiation aux nouveautés dela science 21, Dunod, Paris, (1970).

[2] C. Berge, Graphes et hypergraphes, Dunod, Paris, (1970).

[3] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, North-Holland, New-York, (1976).

[4] B. Bollobàs, Graph Theory, An Introductory Course, Graduate Text in Math. 63, Springer, (1979).

[5] N. Biggs, Algebraic Graph Theory, 2nd Edition, Cambridge Math. Library, (1993).

[6] J.-P. Allouche, J. Shallit, Automatic sequences, Theory, Applications, Generalizations, Cambridge Univ. Press, Cambridge (2004).

[7] V. Bouchitté, Graphes et algorithmique des graphes, notes de cours, Ecole normale supérieure de Lyon.

[8] R. Diestel, Graph Theory, 3rd Edition, Graduate Text in Math. 173, Springer, (2005).

[9] F.R. Gantmacher, The theory of matrices, Chelsea, (1960).

[10] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, Cambridge University Press, (1994).

[11] C. Godsil, G. Royle, Algebraic Graph Theory, Graduate Text in Math. 207, Springer,(2001).

[12] T. Harju, Lecture Notes on Graph Theory, University of Turku, Spring 2002.

[13] B.W. Kernighan, D.M. Ritchie, The C Programming Language, 2nd Ed., Prentice Hall, (1988).

[14] D. Lind, B. Marcus, Symbolic Dynamics and Coding, An Introduction, Cambridge University Press, (1995).

[15] P. Lopez, Cours de GRAPHES, notes de cours (2005).
www.laas.fr~lopez/cours/GRAPHES/graphes.html.

[16] K. Loudon, Maitrise des algorithmes en C, Ed. française, O'Reilly, Paris, (2000)

[17] C. Meyer, A. Langville, Updating the PageRank Vector, SIAM Annual Meeting, New Orleans, 2005.

[18] C. Meyer, A. Langville, Google's PageRank and Beyond: The Science of Search Engine Rankings, Princeton University Press, (2006).

[19] Ministère de l’´éducation nationale, Introduction d’´éléments de la théorie des graphes, accompagnent de la mise en œuvre des programmes, (2001).

[20] O. Ore, Graphs and their uses, version révisée par R. J. Wilson, New Mathematical Library 34, Math. Association of America.

[21] J. Oxley, On the interplay between graphs and matroids, Surveys in combinatorics,

2001 (Sussex), 199–239, London Math. Soc. Lecture Note Ser. 288, Cambridge Univ. Press, (2001).

[22] L. Page, S. Brin, R. Motwani, T.Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical Report, Stanford University, 1998.

[23] J. Miles Prystowsky, L. Gill, Calculating Web Page Authority Using the PageRankAlgorithm, http://online.redwoods.cc.ca.us/instruct/darnold/LAPROJ/.

[24] S. P. Radziszowski, Small Ramsey Number, Dynamic surveys, The Electronic Journal of Combinatorics, http://www.combinatorics.org/Surveys/ds1.pdf