Chapitre II.

Graphe, Arbre couvrant et algorithmes pour TSP

Rappel. Graphe

Un graphe $G=(V,E)$ est un couple (V,E) où
- V : est l'ensemble des sommets
- E : l'ensemble des arêtes ou arcs.

(1) un graphe soit qu'il est orienté ou NON

cas orienté :

Un graphe orienté est un couple $G(V,E)$, où
- $V(G)$: est un ensemble appelé sommets
- $E(G)$ est un ensemble d'arcs.

- Si $a=(x\,y)$ est un arc alors origine$(a)=x$ et destination$(a)=y$
- on peut définir les opérations successeurs(S) et predecesseur(S). respect. l'ensemble des sommets successeurs d'un sommet (S) et l'ensemble de predecesseur de S.
- on peut définir deux autres opérations Sortant(S) et entrant(S). respect. l'ensemble des arcs sortants d'un sommet S. et entrants du sommet S.
- $\delta^-(S)$: degré entrant d'un sommet S est le nombre d'arcs entrant de de S.
- $\delta^+(S)$ degré sortant est le nombre d'arcs sortant de S.
- degré $\delta(S)=\delta^-(S)+\delta^+(S)$

b) - graphe non orienté

un graphe non orienté $G(V, E)$ est un couple :
$V(G)$: un ensemble de sommets.
$E(G)$: " d'arêts.

- Si a est un arc composé des extrimités x et y, alors les deux sommets x et y sont adjacents.

- l'arc a est dite incidente à x si x est l'une des extrimités de a. alors $\delta(x)$ degré lié au sommet x est $\delta(x) = \text{Card}(\text{incid}(x))$.

(2)

Sous-graphe :

- un sous-graphe de G est un graphe de H tel que $V(H) \subseteq V(G)$ et $E(H) \subseteq E(G)$. on note $H \subseteq G$.
un sous-graphe de G est dit **couvrant** si son [redacted] de sommets est $V(G)$

- Si $X \subseteq V(G)$, le sous-graphe induit par X est le sous graphe H de G tel que :
$$V(H) = X \text{ et } E(H) = \{(u,v) \in E(G) \mid u, v \in X\}.$$

- Si $Y \subseteq E(G)$, le sous-graphe induit par Y est le sous-graphe H de G tel que :
$E(H) = Y$ et $V(H)$ est l'ensemble des extrimités des arcs de Y.

Chemins, circuits & cycles :-

Un chemin dans un graphe orienté (resp. non orienté) G est une suite finie $p = x_0 a_1 x_1 a_2 \cdots x_{k-1} a_k x_k$ tel que pour tout i : $x_i \in V$, $a_i \in E$ et $a_i = (x_i, x_{i+1})$. Les sommets x_0 et x_k sont les extrimités de p. le nombre d'arcs (resp. d'arêts) par lesquelles passe le chemin est appelé sa longueur.

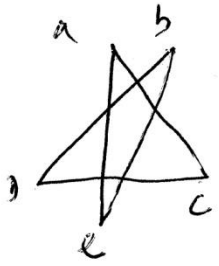
- un chemin est dit élémentaire si, en dehors de ses extrémités, les sommets par lesquels il passe sont deux à deux distincts.
- un chemin dont les extrémités coïncident est appelé circuit dans un graphe non orienté, un tel chemin est appelé cycle
- un graphe non orienté sans cycle est dit acyclique.

Arbres :

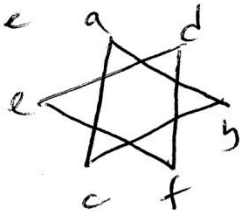
Connexité :

un graphe non orienté est connexe si pour tout sommets x et y il existe une chaine reliant x à y

exple



Connexe



deux composante connexe

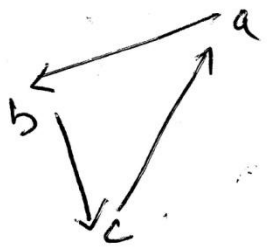
- les composantes connexes d'un graphe $G(V,E)$ forment une partition de V.

- un graphe est connexe SSI il a une seule composante connexe.

Fortement connexe :

Un graphe orienté est fortement connexe si pour tout couple de sommets x et y il existe un chemin reliant x à y

ex.



- un chemin de (a) à (c)
- " de (c) à (a)

déf arbre :

- Un graphe connexe sans cycle est appelé un arbre

- Un graphe connexe sans cycle appelée une forêt. Chacune de ses composantes connexes est un arbre.

Autre courant mathématique :

(4) 1

Cycle et coupes : définitions

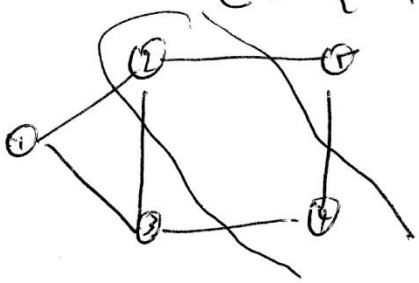
- une coupe est la partition des sommets en deux sous-ensembles, on appelle aussi coupe, l'ensemble des arêtes ayant une extrimité dans chaque sous ensemble de la partition

Coupe minimum : est une coupe contenant un nombre minimum d'arêtes (problème polynomial)
Contrairement au problème de coupe maximum qui est NP-difficile

Coupe maximum : une coupe contenant autant d'arêtes que n'importe quelle autre coupe.
Une coupe est maximum si son poids est maximum parmi tout les coupes

ex

$C = \{1, 3, 5\}$.



Arbre - Couvrant minimal

dans un graphe non-orienté (pondéré), un arbre Couvrant minimale A est un ensemble d'arêtes de G tel que;

1/- est un arbre (G connexe sans cycle c-à-d acyclique)

2/ touche tous les Sommets.

3/ est de poids minimale - c-à-d/

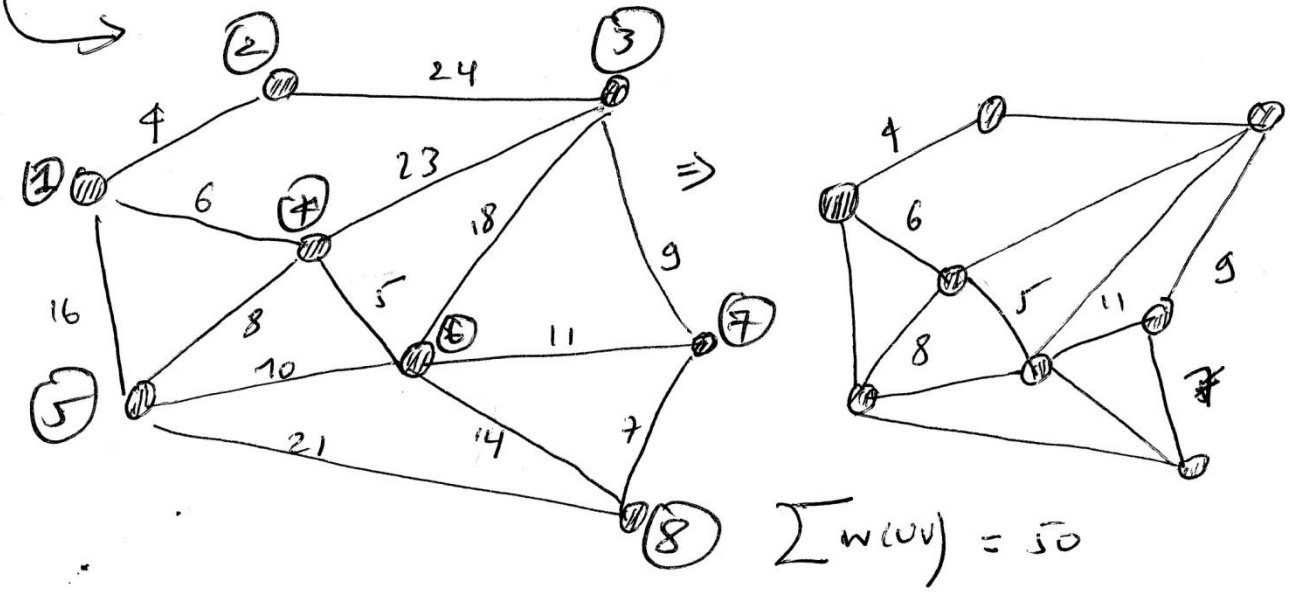
$$\sum_{uv \in A} W(u,v) \text{ est minimal.}$$

exple:

Algorithme de kruskal

capable d'avoir une forêt couverant F de G

F est visible initialement, en ajoutant une à une les arrêtes de poids minimum.



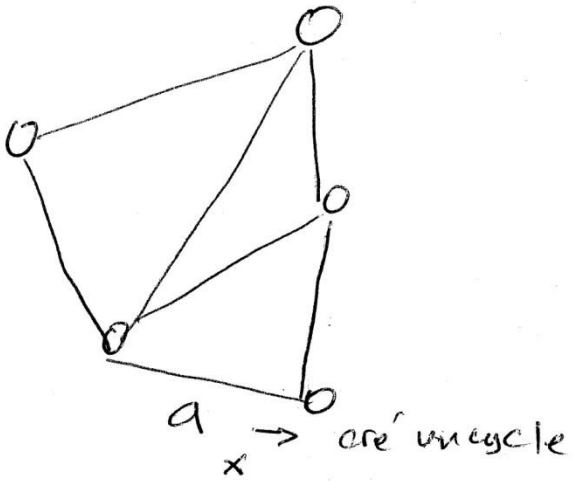
$\sum w(uv) = 50$

Algorithme de Kruskal :

le principe est la construction d'une arbre F. Initialement $F = \{\}$. on considère les arêtes selon l'ordre croissant des poids. on ajoute l'arête a à F si elle n'introduit pas de cycle.

sinon, on ajoute a à F

ex :



Algorithme Kruskal // F est une couverture Minimal Minimum Spanning tree.

début

trier les arêtes selon leur poids Ascendant

$c_1 \leq c_2 \leq \dots \leq c_m$ m : arêtes.

$F \leftarrow \phi$; $P \leftarrow \phi$;

pour tout Sommet $s \in V$ faire

$P \leftarrow P \cup \{s\}$ // initaliser P par des ensembles disjoints de Sommets

. pour $i = 1 \text{ — } m$ faire { m ensemble d'arêtes trier dans l'ordre }

$e_i \leftarrow (s,t)$; arc

si (s et t) sont dans deux ensembles ≠ alors

// si find(P,s) ≠ find(P,t))

$F \leftarrow F \cup \{e_i\}$;

fusionner dans P les deux ensembles contenant s et t

finpour

Retourner F

fin

Algorithme de prim

le principe est le suivant :

- Initialiser un ensemble $S = \{ s \}$, s un sommet quelconque
- on applique la propriété de la coupe à S
- On ajoute à F l'arête de plus petit poids de la coupe correspondant à S. et on ajoute un nouveau sommet exploré v à S

```
Algorithme prim (G: Graphe);
début
    // F est la couverture selon prim()
    F ← { s }   // avec s : sommet quelconque
    tantque V[F] ≠ V[G) faire
        // V[F] : ensemble des sommets de l'MST;
        // V[G] : l'ensemble des sommets
        Selectionner une arête (u, v) de plus
        faible poids tq/
            u ∈ V(F) et v ∉ V(F)
        E(F) = E(F) ∪ (u, v);
        V[F] = V[F] ∪ {v}
    fintantque
fin
```

Voyageur de commerce métrique.

- Etant donné un graphe complet muni d'une fonction de coût positive sur les arêtes. trouver un cycle de coût minimum passant par chaque sommet exactement une fois.

Une 2- Approximation facile:

nous utiliserons le coût d'une arbre couvrant minimale de G. l'arbre MST est minimale selon l'algorithme Kruskal.. Ese

- Théorème:

si $P \neq NP$ et soit $\delta \geq 1$, il n'existe aucun algorithme d'approximation en temps polynomial avec un facteur δ pour le TSP

preuve: ?

Tsp métrique: un TSP métrique est un Graphe TSP qui vérifient l'inégalité triangulaires

soit u, v, w trois sommet $\in V$. alors

$$Cout(u,v) \leq Cout(u,w) + Cout(w,v)$$

Approximation de TSP - Métrique:

nous utiliserons le coût déterminé d'un arbre couvrant minimale MST pour G.

Algorithme TSP-métrique <approximation> :

début

1° Construire un arbre couvrant minimal T.

2° Dupliquer toutes les arêtes du MST pour obtenir un graphe eulérien.

3° trouver un cycle eulérien τ de ce graphe

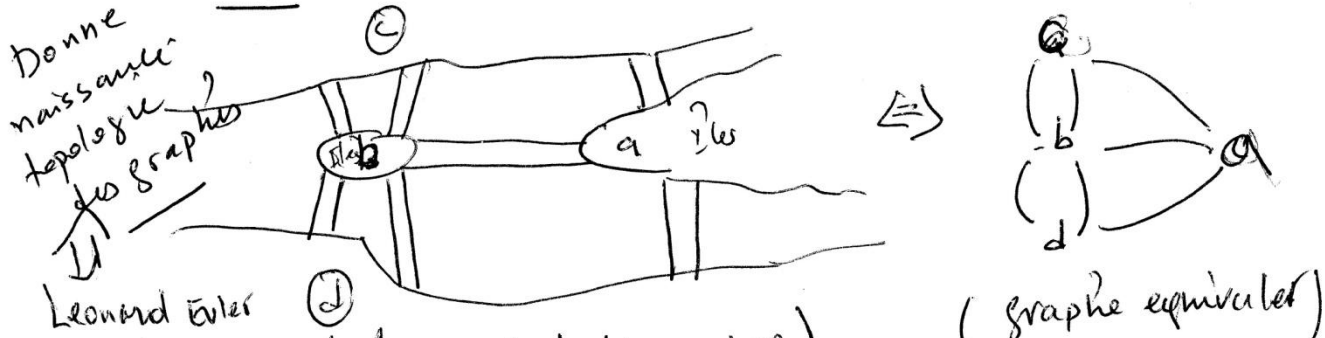
4° Retourner le cycle hamiltonien C de G qui visite les sommets de V dans l'ordre de leur première occurrence dans τ

fin

note : un cycle eulérien est un cycle qui passe une et une seule fois par chaque arête du graphe.

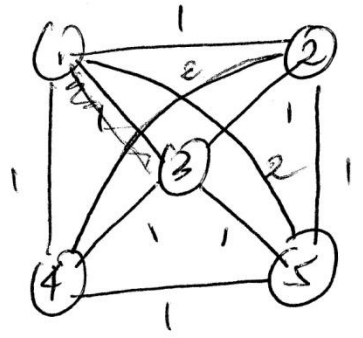
théorème : Un graphe est eulérien ssi il est connexe et tous ses sommets sont de degré pair. [Euler 1736]

exple les ponts de "Koenigsberg" / Kaliningrad Actuellement



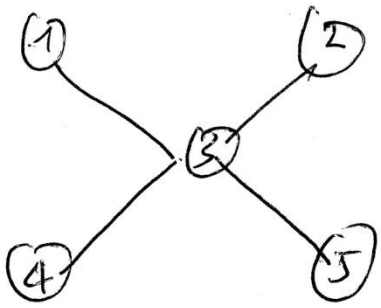
Q/ : peut on faire un promenade dans les rues, à partir d'un point départ

exemple: Soit le graphe $G(V, E)$ dont les arêtes respectent l'inegalité triangulaire. ~~ato.~~

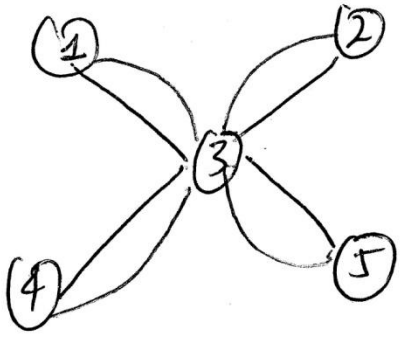


$c(u,v) \leq c(u,w) + c(w,v)$

L'MST



cycle Eulerien après Doublement



= 1 3 2 3 5 3 4 3 1.

alors le cycle Hamiltonien d'après l'ordre d'apparition dans C. est

1 3 2 5 4 1.

théorème:

l'algorithme "tsp-métrique" est une 2-approximation pour le tsp-métrique.

preuve:

Soit opt la solution optimale pour le tsp métrique

alors $Cout(F)$ { F: arbre couvrant MST }.

$Cout(F) \leq OPT$. (1)

- $Cout(C) = 2\,cout(F)$ puisque C contient chaque arête de F en double.
- Alors $Cout(C) \leq Cout(C)$.

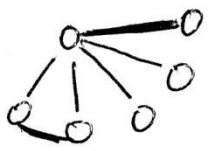
$\Rightarrow Cout(C) \leq 2\,Cout(F)$ (2)

$\Rightarrow Cout(C) \leq 2\,OPT$.

Amélioration de 2-approximation Algorithme:

- déf: un couplage d'un graphe est un ensemble d'arêtes de ce graphe qui n'ont pas de sommets en commun

ex:



- un couplage parfait ou couplage complet est un couplage M du graphe tel que tout sommet du graphe est incident à exactement une arête de M

ex: tout sommet $s \in V$ est extrémité de l'une des arêtes du graphe G.

.. Rappelons qu'un graphe admet un cycle eulerien ssi tous ses sommets sont de degré pair. Alors nous devons nous occuper que des sommets de degré impair du MST.

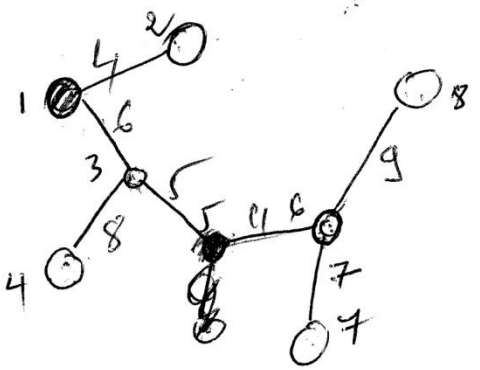
Soit V' l'ensemble de ses sommets. Alors il n'existe qu'un nombre pair des sommets de degré impair car la somme des degré est pair

$$\sum \text{degré} = 2k = \underbrace{2k'}_{\text{pair}} + \underbrace{m}_{\substack{\text{degré impair} \\ \text{dans MST}}}$$

$$\Rightarrow m = 2(k - k') = 2m' \text{ alors le nombre est pair.}$$

$$\Rightarrow |V'| = 2m' \text{ pair.}$$

ex

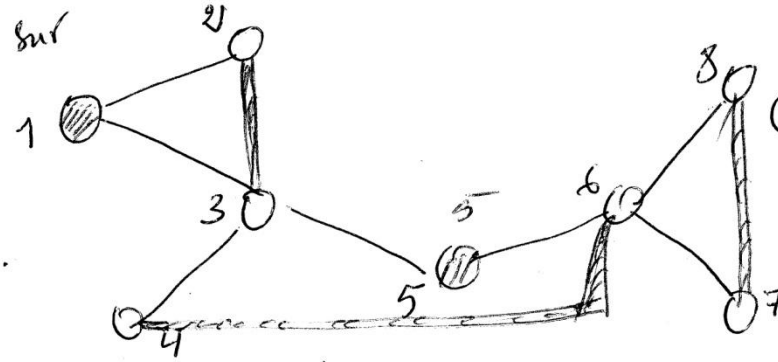


$|V'| = 6$. $V' = \{2, 3, 4, 6, 7, 8\}$.

~~le nombre de sommet~~ Somme

- la somme de degré des sommets d'un graphe est pair (le double du nombre d'arêtes).

Ajout couplage sur V'



(*) Alors tous les sommets sont de degré pair