

Problème du postier et du voyageur de commerce ; algorithmes d'approximation

CM n°9 — Algorithmique (AL5)

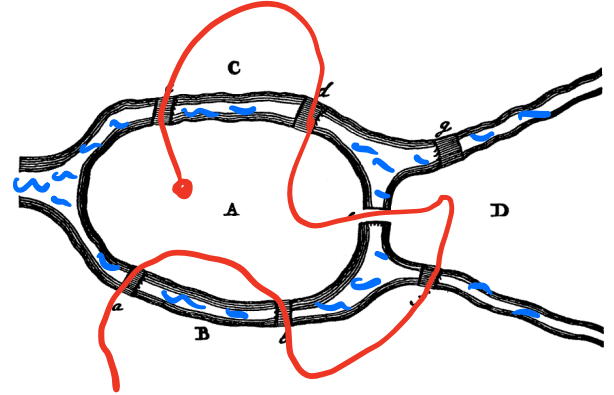
Matěj Stehlík

24/11/2023

Les sept ponts de Königsberg

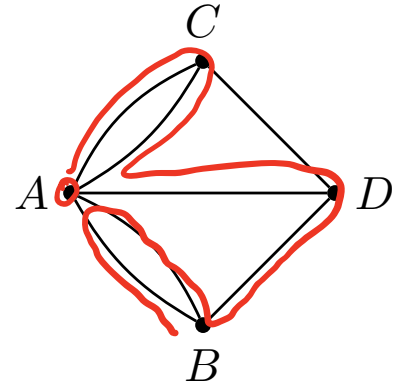
Problème

Existe-t-il une promenade dans les rues de Königsberg permettant, à partir d'un point de départ au choix, de passer une et une seule fois par chaque pont, et de revenir à son point de départ ?



Une reformulation

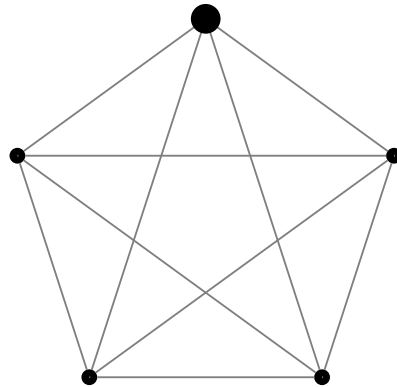
Existe-t-il un cycle dans le graphe à droite qui traverse chaque arête exactement une fois ?



Graphes eulériens

Définition

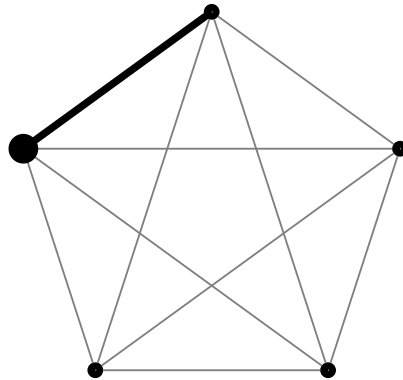
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

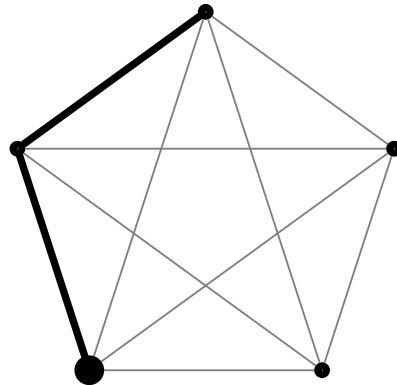
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

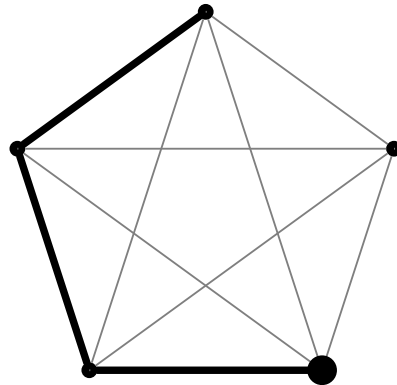
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

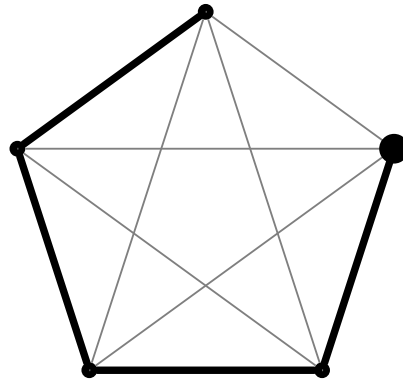
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

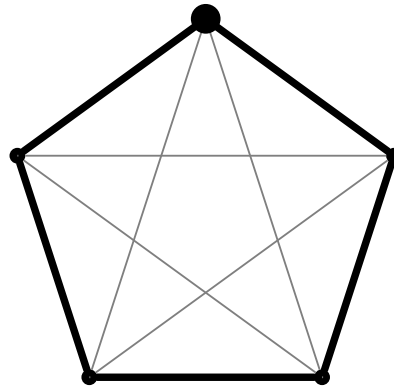
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

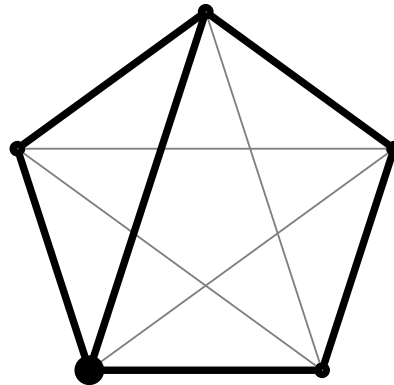
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

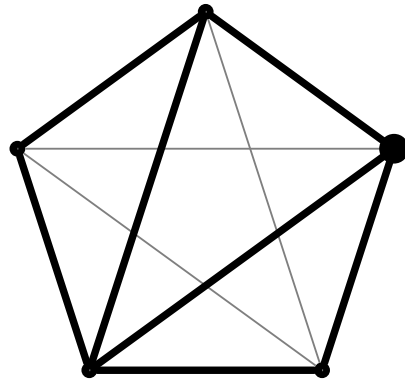
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

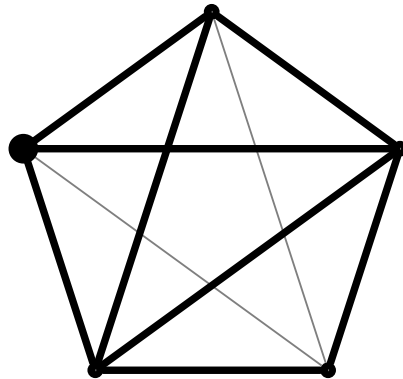
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

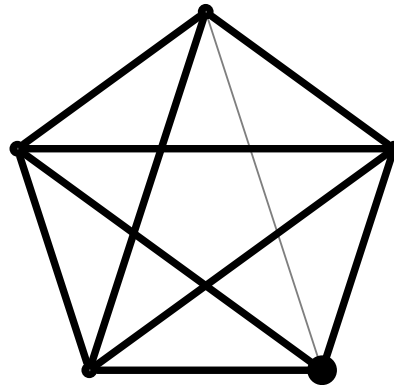
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

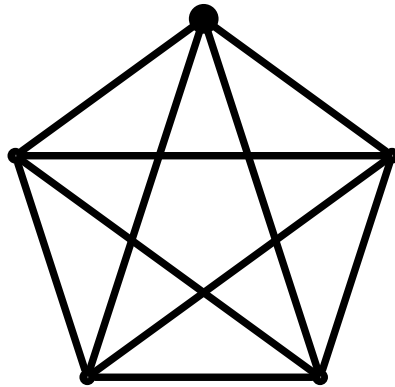
- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



Graphes eulériens

Définition

- Un cycle C dans un graphe G est *eulérien* si C traverse chaque arête de G une et une seule fois.
- Un graphe avec un cycle eulérien est appelé *graphe eulérien*.



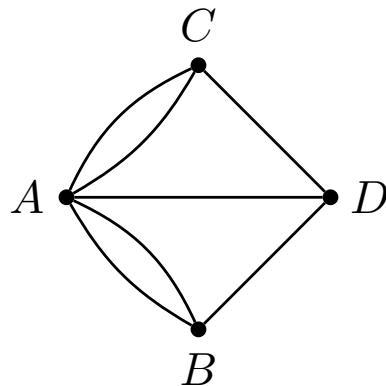
Caractérisation des graphes eulériens

Théorème (Euler 1736)

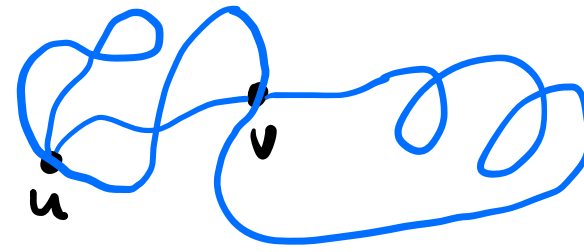
Un graphe G est eulérien si et seulement si G est connexe, et tout sommet de G est de degré pair.

Remarques

- Euler a prouvé eulérien \implies connexe et tous les degrés pairs
- Il a ainsi donné une réponse négative au problème des sept ponts.



Démonstration du théorème d'Euler



(\Rightarrow)

- Soit G un graphe eulérien avec un cycle eulérien C avec sommet initial et terminal u .
- Chaque fois que C passe par un sommet $v \neq u$, on compte deux arêtes incidentes à v .
- Donc, le degré de v est pair.
- De même, $d(u)$ est pair puisque C débute et termine en u .

(\Leftarrow)

- Soit G un graphe connexe dont tous les sommets sont de degré pair.
- On peut construire un cycle eulérien en utilisant l'algorithme de Hierholzer.

L'algorithme de Hierholzer

$$\sum d(v) = \text{nombre pair}$$

\Rightarrow nombre de sommets

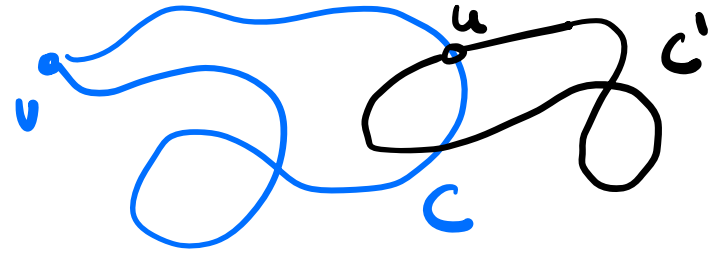
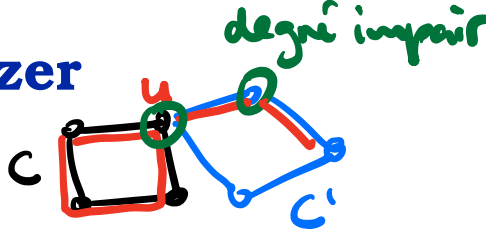
de deg
impair

est pair

- Choisir n'importe quel sommet initial v
- Suivre un chemin arbitraire d'arcs jusqu'à retourner à v , obtenant ainsi un cycle C .
- Tant qu'il y a des sommets u dans le cycle C avec des arcs qu'on n'a pas encore choisis faire
 - Suivre un chemin à partir de u , n'utilisant que des arcs pas encore choisis, jusqu'à retourner à u , obtenant un cycle C'
 - Prolonger le cycle C par C'

- La complexité de l'algorithme est de $O(m)$

Lorsqu'on entre un sommet de degré pair, par une arête, le nombre d'arêtes non traversées incidentes au sommet est impair, et donc ≥ 1 .



(Puisque G est connexe, on a $m \geq n-1$, donc $n = O(m)$, donc $O(n+m) = O(m)$.)

Chaînes eulériennes

- Une chaîne P dans un graphe est *eulérienne* si P traverse chaque arête du graphe une et une seule fois.
- Une conséquence facile du théorème d'Euler :

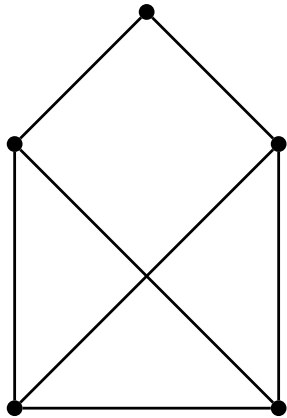
Théorème

Un graphe G contient une chaîne eulérienne ssi G est connexe et G contient exactement 2 sommets de degré impair (tous les autres sommets sont de degré pair).

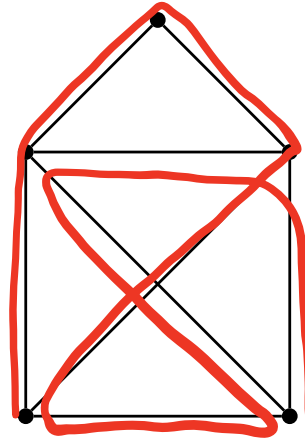
Dessiner sans lever le crayon

Corollaire

On peut dessiner un graphe G sans lever le crayon ssi G a au plus deux sommets de degré impair.



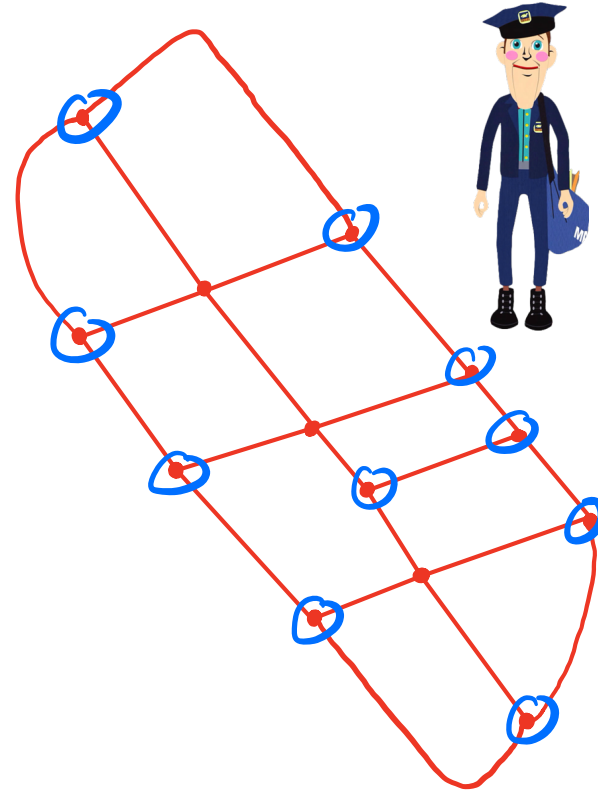
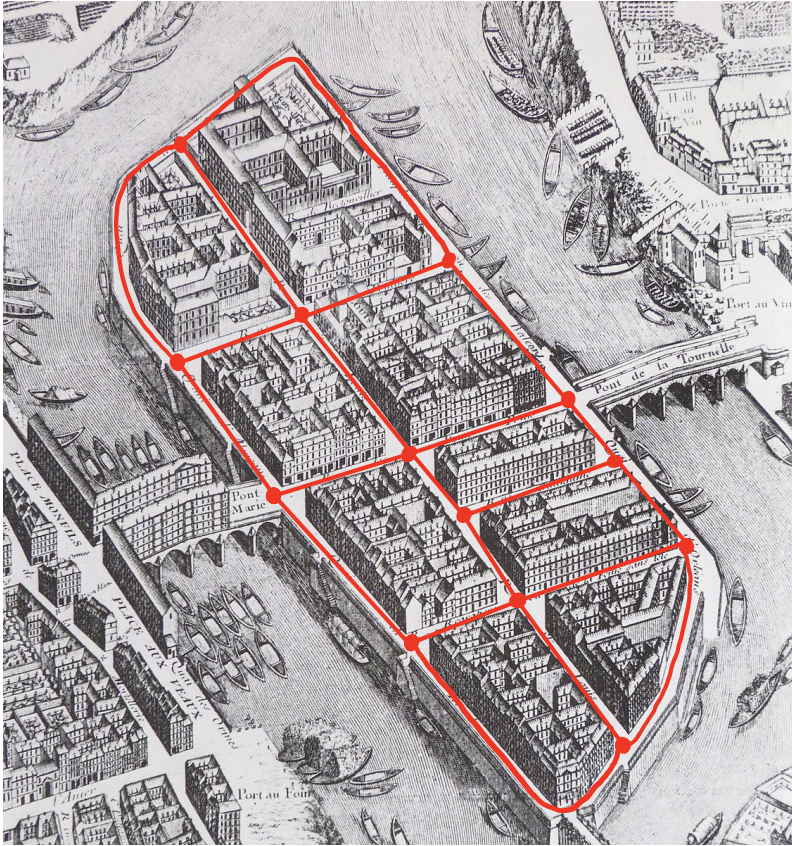
pas possible



Le problème du postier chinois

- Un postier veut livrer le courrier le long de toutes les arêtes d'un graphe $G = (V, E)$ et revenir à son point de départ.
- Un tel chemin est appelé un *tour de postier chinois* de G .
- Il peut être nécessaire de parcourir quelques arêtes plusieurs fois.
- Soit $c_e \geq 0$ le coût associé à chaque traversée d'une telle arête e .
- Le *problème du postier chinois* : trouver un tour de postier chinois de coût minimum.
- Si le graphe G contient un cycle eulérien C , alors C est forcément un tour de postier optimal.

Une illustration



Quelques applications du problème du postier chinois

- Planification de l'entretien des rues
- Ramassage des poubelles
- Acheminement des chasse-neige
- Vérification des hyperliens d'un site web (version orientée)

Le problème du postier chinois

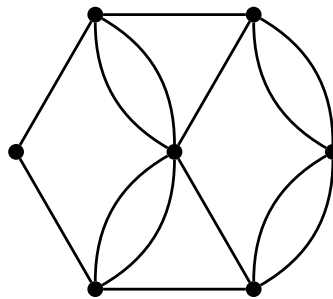
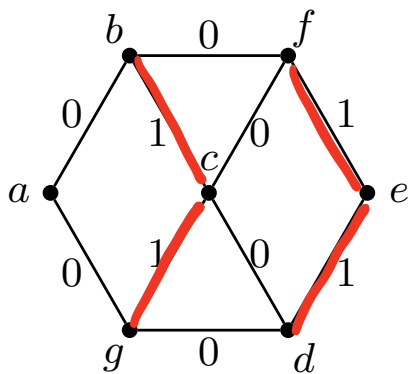
Problème du postier chinois

Entrée Un graphe $G = (V, E)$ avec pondération $c \in \mathbb{R}^{|E|}$ tel que $c \geq 0$.

Objectif Trouver un plus court tour passant au moins une fois par chaque arête de G .

Le problème du postier chinois

- Supposons qu'on a un tour de postier chinois dans G .
- Soit x_e le nombre de traversées supplémentaires de l'arête e , pour toute $e \in E$.
- Soit G^x le supergraphe de G avec $1 + x_e$ copies de l'arête e , pour toute $e \in E$.



un ensemble de postier J

Le problème du postier chinois

- Il est facile à voir que G^x a un cycle eulérien.
- Supposons que $x \in \mathbb{Z}^E$ est non négatif, avec la propriété que G^x ait un cycle eulérien.
- Puis il existe un tour de postier chinois de G qui emprunte chaque arête e $1 + x_e$ fois.
- Au vu de ces observations et le théorème d'Euler, on peut reformuler le problème de la façon suivante :

$$\min \sum_{e \in E} c_e x_e$$

$$\text{s.c.} \quad x(\delta(v)) \equiv d(v) \pmod{2} \quad \text{pour tout } v \in V$$

$$x_e \geq 0 \quad \text{pour toute } e \in E$$

$$x_e \in \mathbb{N}$$

Reformulation du problème du postier chinois

- Il y a une solution optimale où x prend les valeurs dans $\{0, 1\}$: si $x_e \geq 2$, on peut diminuer x_e de deux, ce qui donne une autre solution de coût inférieur ou égal à celui de la solution originale.
- Il suffit donc chercher un ensemble J d'arêtes (avec vecteur caractéristique x).
- J est un *ensemble de postier* de G si pour tout sommet $v \in V$, v est incident à un nombre impair d'arêtes dans J ssi v est de degré impair dans G .
- Le problème peut être reformulé comme suit :

Problème du postier chinois

Entrée Un graphe $G = (V, E)$ avec pondération $c \in \mathbb{R}^{|E|}$ tel que $c \geq 0$.

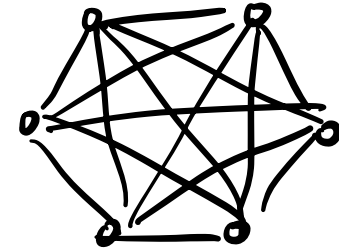
Objectif Trouver un ensemble postier J tel que $c(J)$ soit minimum.

Résolution du problème du postier chinois

- Soit $d(u, v)$ le coût d'une (u, v) -chaîne de coût minimum dans G .
- Soit $|T| = 2k$.
- Le coût minimum d'un ensemble de postier est :

$$\min \sum_{i=1}^k d(u_i, v_i)$$

\hat{G}



s.c. u_1v_1, \dots, u_kv_k est un jumelage (couplage) des éléments de T .

- C'est un problème de couplages !
- On forme un graphe complet $\hat{G} = (T, \hat{E})$, on affecte le poids $d(u, v)$ à chaque arête uv , et on trouve un couplage parfait de poids minimum.
- Ceci détermine un jumelage des sommets de T , et donc un ensemble de chaînes qui rejoignent les sommets jumeaux.

Algorithme de postier chinois

Entrée : Graphe $G = (V, E)$ avec une pondération $c \in \mathbb{R}^{|E|}$ t.q. $c \geq 0$.

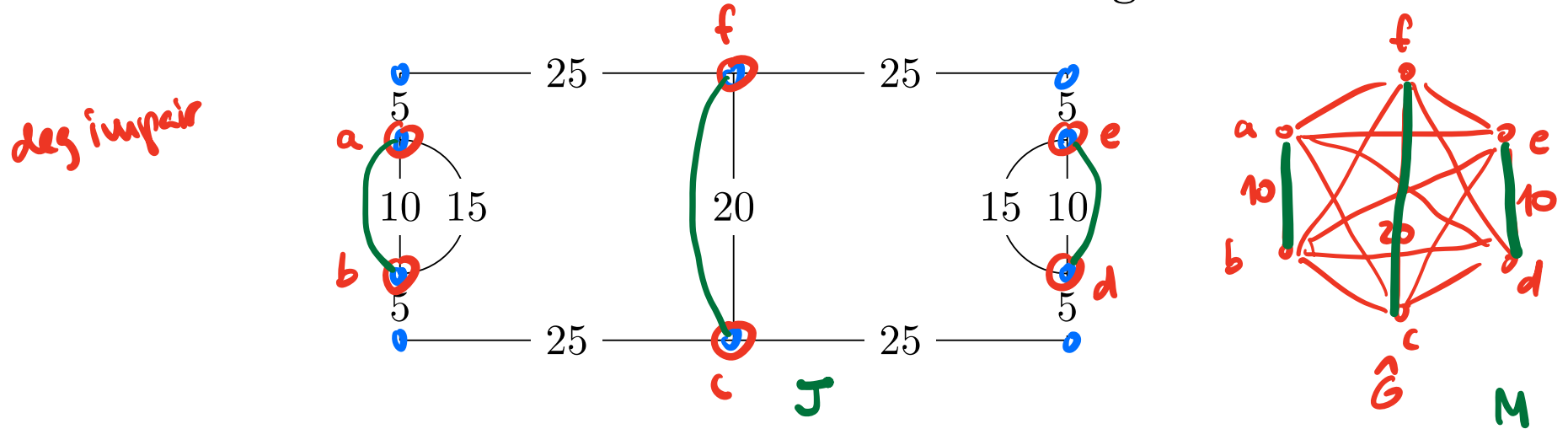
Sortie : Un ensemble de postier $J \subseteq E$.

1. Pour chaque paire $\{u, v\}$ de sommets, trouver un (u, v) -chemin P_{uv} de coût minimum. Soit $d(u, v)$ le coût (longueur) de P_{uv} .
2. Soit $\hat{G} = (T, \hat{E})$ un graphe complet où l'arête uv est de poids $d(u, v)$, pour toute arête $uv \in \hat{E}$. Trouver un couplage parfait M de poids minimum dans G . *$T = \{\text{sommets de } G \text{ de degré impair}\}$*
3. Soit J la différence symétrique des arêtes des chemins P_{uv} pour $uv \in M$.

La complexité de cet algorithme est de $O(n^4)$, dominé par l'étape 2 (algorithme d'Edmonds).

Exemple

Le jardinier doit marquer les lignes blanches sur un terrain de foot à 5, dont les distances (en mètres) sont données dans le diagramme suivant.



L'appareil à marquer les lignes étant cassé, il ne peut pas être arrêté et marque des lignes partout où il passe.

Question

Quelle distance doit parcourir le jardinier?

$$\begin{aligned} &\sum \text{ poids des arêtes} \\ &+ 10 + 20 + 10 \end{aligned}$$

coupage de poids min

Problèmes d'optimisation “durs”

- Un *problème d'optimisation* est le problème de trouver la meilleure solution parmi toutes les solutions réalisables.
- Nous avons déjà vu plusieurs exemples :
 - plus court chemin
 - arbre couvrant de poids minimum
 - flot maximum
 - coupe minimum
 - couplage maximum
 - transversal minimum dans les graphes bipartis
 - tour de postier chinois. . .
- Chacun de ces problèmes peut être résolu par un algorithme en temps *polynomial* (par rapport à la taille de l'instance).
- Pour certains problèmes, tous les algorithmes connus prennent un temps *exponentiel*. . .

Exemple classique d'un problème d'optimisation dur

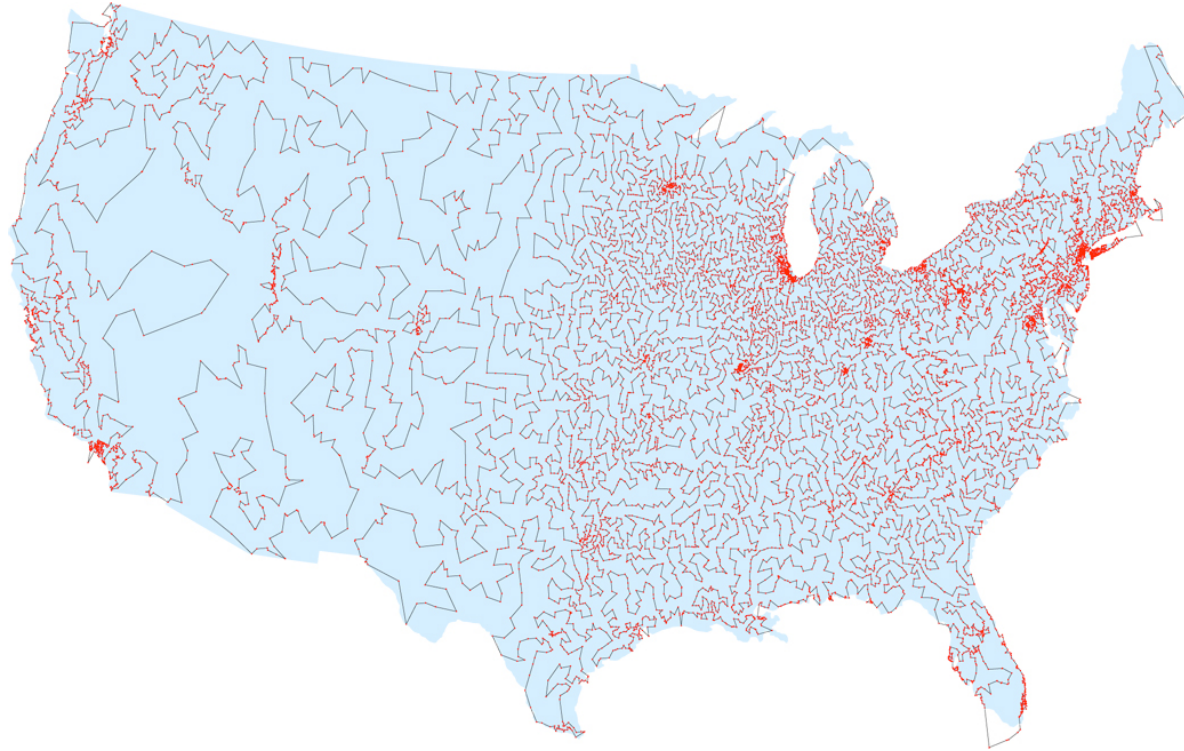
Le problème du voyageur de commerce (TSP)

Instance Un graphe complet $G = (V, E)$ avec pondération $w \in \mathbb{R}^{|E|}$.

Problème Trouver un cycle de poids minimum qui passe par chaque sommet de G une et une seule fois.

- Illustration dans la vie réelle : déterminer, étant donné une liste de villes et les distances entre toutes les paires de villes, le plus court cycle qui passe par chaque ville une et une seule fois.

Exemple d'un tour de voyageur de commerce aux États-Unis



Algorithmes d'approximation

Définition

- Lorsque l'on n'arrive pas à trouver un algorithme polynomial pour un problème d'optimisation, on peut espérer de trouver un *algorithme d'approximation*.
- Étant donné une problème d'optimisation, soit $\text{OPT}(I)$ la solution optimale de l'instance I .
- Soit \mathcal{A} un algorithme (polynomial) qui retourne une solution $\mathcal{A}(I)$ de l'instance I .
- Alors, le *facteur d'approximation* de \mathcal{A} est défini comme

$$\rho = \max_I \frac{\mathcal{A}(I)}{\text{OPT}(I)}.$$

- On dit que \mathcal{A} est un algorithme de ρ -approximation.

Un algorithme d'approximation pour VERTEX COVER

= transversal $(\tau(\dots))$

Algorithme : Vertex-Cover-Approx

Entrées : un graphe $G = (V, E)$

Sorties : un transversal (vertex cover) T de G

début

$T \leftarrow \emptyset$

$F \leftarrow E$

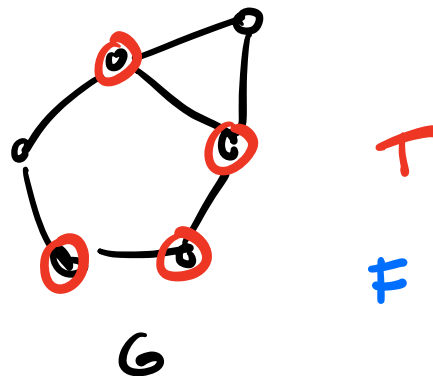
tant que $F \neq \emptyset$ **faire**

 Soit uv une arête de F

$T \leftarrow T \cup \{u, v\}$

$F \leftarrow F \setminus \{\delta(u) \cup \delta(v)\}$

retourner T



Analyse de l'algorithme

Théorème

Vertex-Cover-Approx est un algorithme de 2-approximation pour VERTEX COVER.

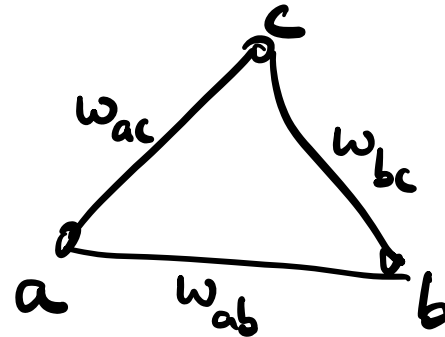
Démonstration

- Soit T la sortie de l'algorithme.
- Il est clair que T est un transversal de G .
- Soient $M = \{u_1v_1, u_2v_2, \dots, u_kv_k\}$ les arêtes choisies par l'algorithme.
- Les arêtes dans M sont disjointes.
- Soit T' un transversal minimal de G .
- T' doit intersecter toutes les arêtes de M , donc $|T'| \geq |M| = k = |T|/2$.
- On conclut que $|T| \leq 2|T'|$.

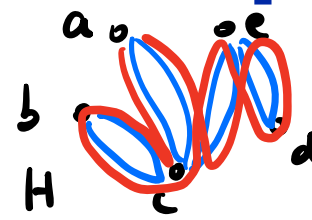
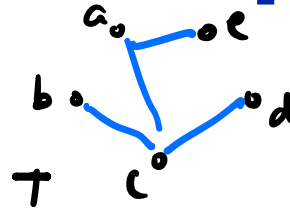
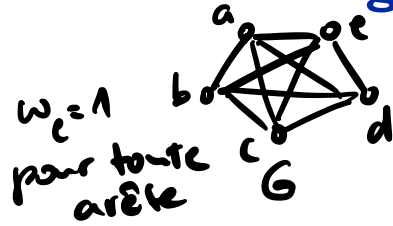
$$\tau(G) \geq \nu(G)$$

Algorithme d'approximation pour le voyageur de commerce... ?

- Aucun algorithme polynomial n'est connu pour résoudre TSP en général.
- Cependant, il existe des algorithmes d'approximation lorsque les poids sur les arêtes satisfont l'inégalité triangulaire : soient a , b et c des sommets de G quelconques, alors $w_{ac} \leq w_{ab} + w_{bc}$.
- On parle dans ce cas du problème de voyageur de commerce (TSP) *métrique*.

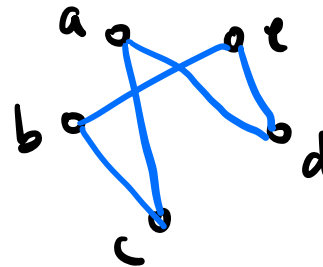


Double-Tree : un algorithme d'approximation pour le TSP métrique



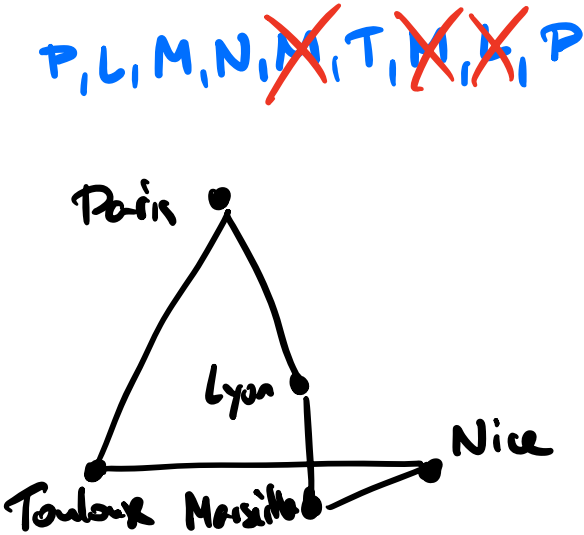
Cycle eulérien :
~~a, c, b, e, d, e, a~~

1. Trouver un arbre couvrant T de poids minimum dans $G = (V, E)$.
2. Construire le multigraphe H avec ensemble de sommets V et deux copies de chaque arête de T .
3. Trouver un cycle eulérien $C = (v_1, e_1, v_2, e_2, \dots, v_{n-1}, e_{n-1}, v_n, e_n, v_1)$ dans H .
4. Prendre les sommets de C dans l'ordre et supprimer tout sauf la première occurrence de chaque sommet (en gardant aussi la dernière occurrence de v_1).



Cycle trouvé

Illustration de l'algorithme Double-Tree



	Paris	Marseille	Lyon	Toulouse	Nice
Paris		777 km	473 km	678 km	930 km
Marseille	777 km		317 km	409 km	201 km
Lyon	473 km	317 km		539 km	470 km
Toulouse	678 km	409 km	539 km		560 km
Nice	930 km	201 km	470 km	560 km	

Complexité de l'algorithme Double-Tree

Remarque

La complexité de l'algorithme Double-Tree est de $O(n^2 \log n)$.

1. trouver un arbre couvrant T de poids minimum : $O(n^2 \log n)$ (Kruskal)
2. construire le multigraphe H : $O(n)$
3. trouver un cycle eulérien C : $O(n)$ (Hierholzer)
4. supprimer des sommets de C : $O(n)$

Le facteur d'approximation de Double-Tree (1/2)

Theorem

Double-Tree est un algorithme de 2-approximation pour le TSP métrique.

- Soit C^* un plus court tour de voyageur de commerce, de longueur OPT .
- Si l'on supprime une arête quelconque de C^* , on obtient un graphe couvrant connexe et acyclique, c'est-à-dire, un arbre couvrant de poids au plus OPT .
- En particulier, si T est un arbre couvrant de poids minimum, alors $w(T) \leq w(C^*) = \text{OPT}$.
- Donc, $w(H) = 2w(T) \leq 2 \cdot \text{OPT}$.

Le facteur d'approximation de Double-Tree (2/2)

- Supposons que le sommet v_i est supprimé lors de la dernière étape de l'algorithme Double-Tree.
- Soient v_{i-1} son prédécesseur et v_{i+1} son successeur dans le cycle.
- Par l'inégalité triangulaire, $w(v_{i-1}v_{i+1}) \leq w(v_{i-1}v_i) + w(v_iv_{i+1})$.
- Donc, aucune suppression d'un sommet dans la dernière étape ne peut augmenter le poids du cycle.
- En particulier, si C' est le cycle final, alors $w(C') \leq w(C) = w(H) \leq 2 \cdot \text{OPT}$.

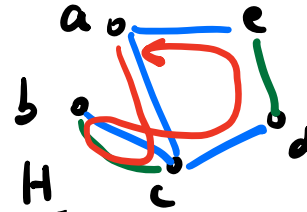
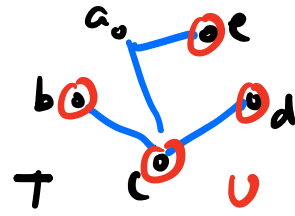
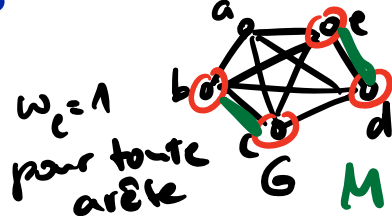
Une amélioration de Double-Tree ?

L'algorithme utilise les idées suivantes :

- construire le multigraphe H eulérien sur $V(G)$ en utilisant les arêtes de G (en se permettant de “dédoubler” des arêtes).
- faire des “raccourcis” pour obtenir un tour de voyageur de commerce

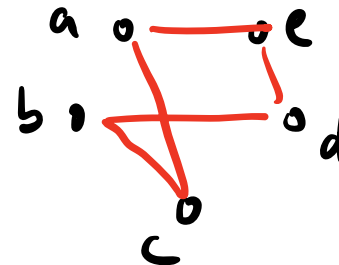
Il y a une façon plus fine de trouver le graphe H .

L'algorithme de Christofides



Cycle eulérien :
 $a, c, b, \cancel{c}, d, e, a$

1. Trouver un arbre couvrant T de poids minimum dans G .
2. Trouver l'ensemble $U \subseteq V$ de sommets de degré impair dans T .
3. Trouver un couplage parfait M de poids minimum dans $G[U]$.
4. Construire un graphe eulérien H en ajoutant les arêtes de M à T .
5. Trouver un cycle eulérien C de H .
6. Prendre les sommets de C dans l'ordre et supprimer tout sauf la première occurrence de chaque sommet (en gardant aussi la dernière occurrence de v_1).



Cycle trouvé

Complexité de l'algorithme de Christofides

Remarque

La complexité de l'algorithme de Christofides est de $O(n^4)$.

1. Trouver un arbre couvrant T de poids minimum dans G : $O(n^2 \log n)$ (Kruskal)
2. Trouver l'ensemble $U \subseteq V$ de sommets de degré impair dans T : $O(nm)$
3. Trouver un couplage parfait M de poids minimum dans $G[U]$: $O(n^4)$ (Edmonds)
4. Construire le graphe H : $O(n)$
5. Trouver un cycle eulérien C : $O(n)$ (Hierholzer)
6. Supprimer des sommets de C : $O(n)$

Le facteur d'approximation de Christofides

Theorem

Christofides est un algorithme de 1.5-approximation pour le TSP métrique.

- Comme dans l'algorithme Double-Tree, on sait que $w(T) \leq \text{OPT}$.
- Il faut démontrer que $w(M) \leq \text{OPT}/2$.
- Soit C^* un cycle hamiltonien dans G de poids OPT .
- Soient u_1, u_2, \dots, u_{2k} les sommets de degré impair dans T , numérotés dans le sens de C^* .
- Soit $M_1 = \{u_1u_2, u_3u_4, \dots, u_{2k-1}u_{2k}\}$ et $M_2 = \{u_2u_3, u_4u_5, \dots, u_{2k-2}u_{2k-1}, u_{2k}u_1\}$.
- Comme $w(M_1) + w(M_2) = \text{OPT}$, on conclut que $w(M_1) \leq \text{OPT}/2$ ou $w(M_2) \leq \text{OPT}/2$.
- En particulier, $w(M) \leq \text{OPT}/2$.

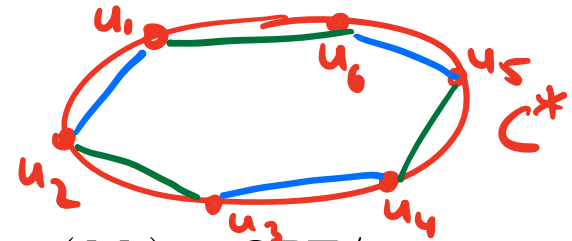
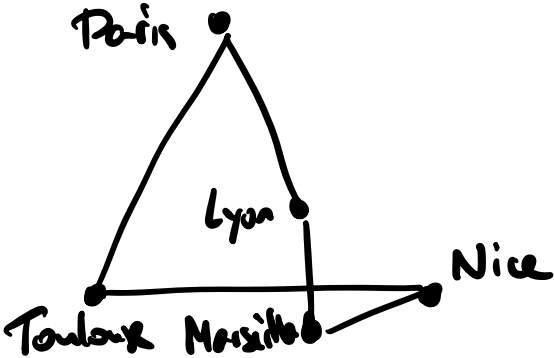


Illustration de l'algorithme de Christofides



P, L, M, N, ~~M~~, T, P



P-T 678
M-N 201

879

	Paris	Marseille	Lyon	Toulouse	Nice
Paris		777 km	473 km	678 km	930 km
Marseille	777 km		317 km	409 km	201 km
Lyon	473 km	317 km		539 km	470 km
Toulouse	678 km	409 km	539 km		560 km
Nice	930 km	201 km	470 km	560 km	