Un algorithme d'approximation pour le TSP métrique

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 $C = (v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}, v_n, e_n, v_1)$ 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 occurence de v_1).

Illustration de l'algorithme

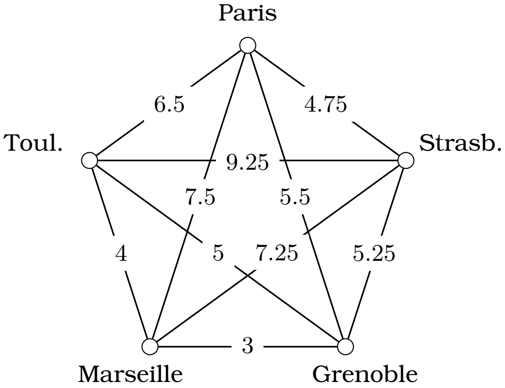
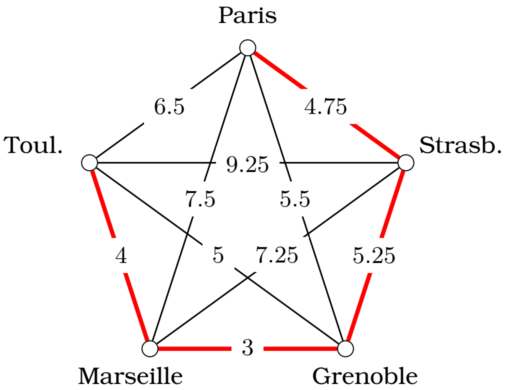


Illustration de l'algorithme



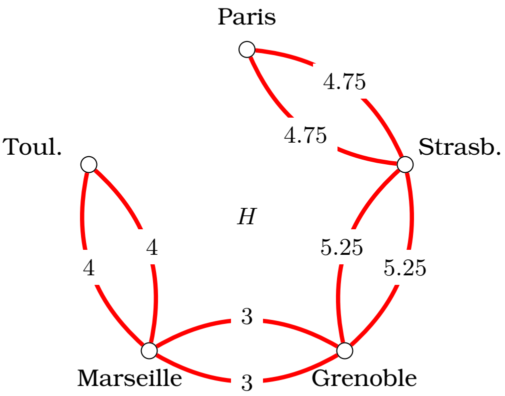
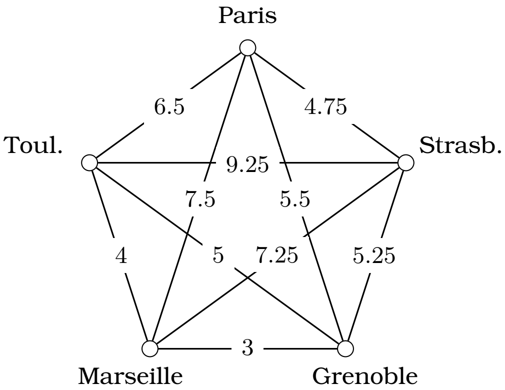
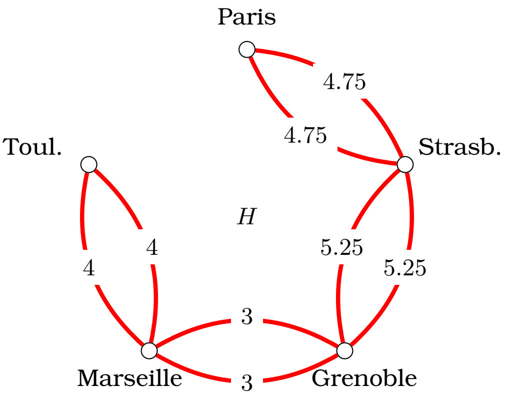


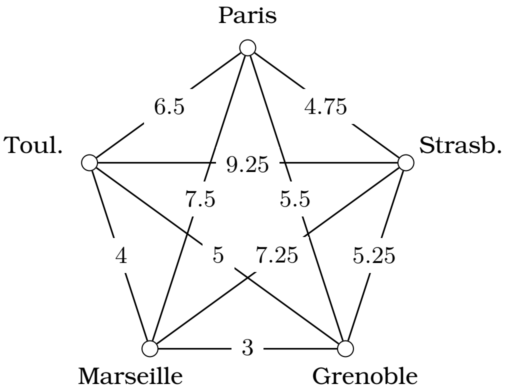
Illustration de l'algorithme

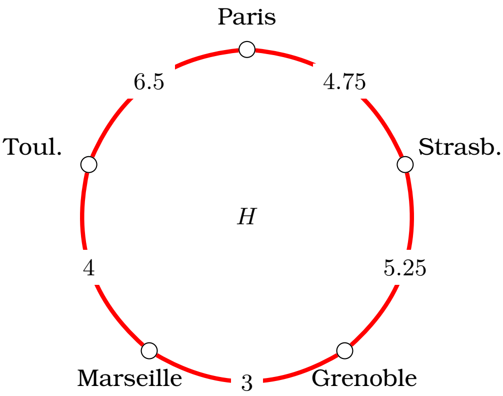




$C = (P, S, G, M, T, M, G, S, P)$

Illustration de l'algorithme





$$C = (P, S, G, M, T, M, G, S, P)$$

Complexité de l'algorithme

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, $w(H) = 2w(T) \leq 2\text{OPT}$.

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 $w(C') \leq w(C) = w(H) \leq 2\mathrm{OPT}$.

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

Entrées : Un graphe complet $G = (V, E)$ avec des poids $w : E \to \mathbb{R}^+$ sur les arêtes qui satisfont l'inégalité triangulaire

Sorties : Cycle hamiltonien $C \subseteq G$ tel que $w(C) \leq 1.5 \cdot \mathrm{OPT}$

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 Faire des “raccourcis” pour obtenir un cycle hamiltonien $C \subseteq G$ (comme dans l'algorithme *Double-Tree*).

Remarques sur l'algorithme de Christofides

- Pourquoi existe-t-il toujours un couplage parfait dans $G[U]$?
- Il existe un algorithme pour trouver un couplage parfait de poids minimum de complexité $O(n^3)$ (pas vu dans ce cours).
- Pourquoi est-ce un algorithme de 1.5-approximation ?

Illustration de l'algorithme

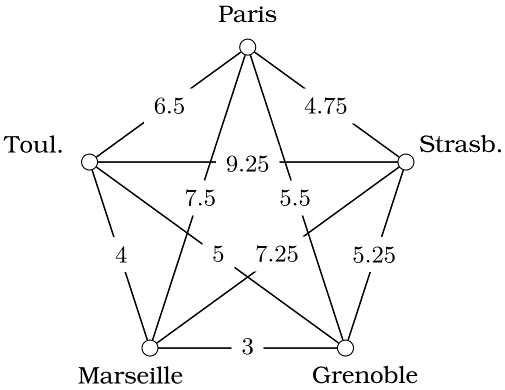


Illustration de l'algorithme