

Algorithmique Avancée Synthèse : étude de TSP

Denis TRYSTRAM
Notes de cours ENSIMAG Alternants 2A

Oct. 2023

Objectif

Revoir la démarche d'analyse des problèmes difficiles à travers un exemple.

Agenda

- Caractérisation et complexité.
- Résolution exacte.
- Heuristiques.
- Algorithme d'approximation.

Le Voyageur de commerce

Présentation informelle

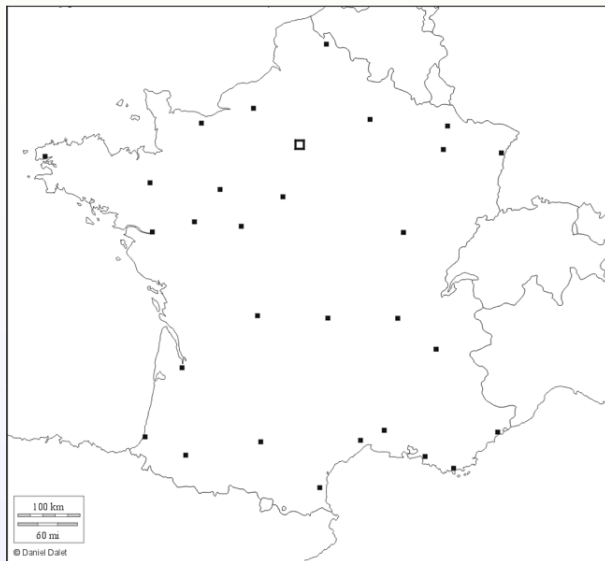
Un voyageur de commerce veut organiser sa tournée saisonnière.

- Il doit passer dans toutes les villes de sa région pour visiter ses clients.
- Il voyage en voiture car il a des chantillons à montrer ces clients, et il veut faire le moins de kilomètres possibles.

Variante : **TSP euclidien** dans le cas où la distance est euclidienne¹.

¹plus court chemin en kilomètre versus chemin le plus rapide.

Une instance (graphique)



Description formelle

- TSP
- **Entrée.** un ensemble V de villes et une matrice des distances inter-villes $(d_{i,j})$.
- **Question.** Déterminer un parcours fermé (passant une fois et une seule par toutes les villes) de longueur minimale.

Remarquons que le codage de l'instance est une matrice.

Codage d'une instance

Codage avec $n = 7$ villes.

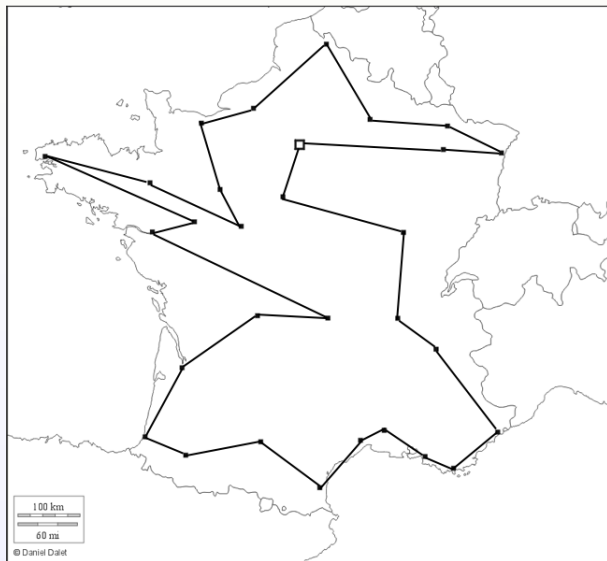
City	Inter-City Costs						
C_1	0	$c_{1,2}$	$c_{1,3}$	$c_{1,4}$	$c_{1,5}$	$c_{1,6}$	$c_{1,7}$
C_2	$c_{2,1}$	0	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$	$c_{2,7}$
C_3	$c_{3,1}$	$c_{3,2}$	0	$c_{3,4}$	$c_{3,5}$	$c_{3,6}$	$c_{3,7}$
C_4	$c_{4,1}$	$c_{4,2}$	$c_{4,3}$	0	$c_{4,5}$	$c_{4,6}$	$c_{4,7}$
C_5	$c_{5,1}$	$c_{5,2}$	$c_{5,3}$	$c_{5,4}$	0	$c_{5,6}$	$c_{5,7}$
C_6	$c_{6,1}$	$c_{6,2}$	$c_{6,3}$	$c_{6,4}$	$c_{6,5}$	0	$c_{6,7}$
C_7	$c_{7,1}$	$c_{7,2}$	$c_{7,3}$	$c_{7,4}$	$c_{7,5}$	$c_{7,6}$	0

version de décision de TSP

Considérons la version de décision du problème du voyageur de commerce (D-TSP).

- D-TSP
- **Entrée.** un ensemble V de villes et une matrice des distances inter-villes $(d_{i,j})$.
Une constante k .
- **Question.** Déterminer s'il existe un parcours fermé, passant une fois et une seule par toutes les villes, de longueur inférieure à k .

Une solution possible



Complexité

D-TSP est NP-complet

La réduction est obtenue à partir de HC (cycle hamiltonien) :

- L'ensemble des villes correspond aux sommets du graphe G
- Les distances sont données par $d_{i,j} = 1$ si i et j sont reliés, 2 sinon.
- La constante k est égale à n (nombre de villes).

Preuve

Cette transformation τ est polynomiale de manière évidente.

- Si G possède un cycle hamiltonien alors ce cycle correspond à une tournée (qui passe par toutes villes une fois et une seule) et sa longueur est n , donc $\tau(G)$ est positive.
- Réciproquement, une instance positive de D-TSP dans $\tau(G)$ est transformée en une instance positive de HC.
En effet, la solution de D-TSP possède par définition des arêtes de coût unitaire puisque la longueur totale est n , ce qui correspond bien à un cycle hamiltonien de G .

Complexité

- La version décision est NP-complet, donc la version "optimisation" est difficile.
- Le temps d'une exploration exhaustive est en $\mathcal{O}(n!)$

Résolution exacte

- Rappels sur la méthode

La première question à se poser :

Que pourrait construire une solution exhaustive ?

- Trouver une stratégie de parcours des solutions
- Peut-on obtenir une bonne solution réalisable ?
- Trouver une borne inférieure efficace.

TPS est un problème de *min*.

Séparation

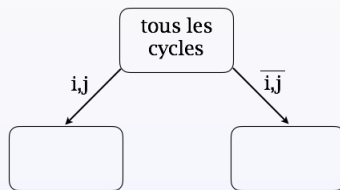
Petit brain storming sur les possibilités de séparation.



Une première méthode

Algorithme de Little

On part ici sur une séparation binaire où l'on distingue si une arête est retenue ou non.



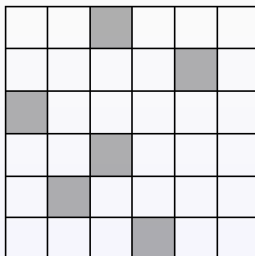
Nous allons privilégier la branche de gauche.

Comment évaluer ?

Considérons un exemple avec 6 villes.

∞	90	30	40	120	50
90	∞	80	60	20	100
30	80	∞	40	90	80
40	60	40	∞	70	50
120	20	90	70	∞	70
50	100	80	50	70	∞

Réduction (en minimum)



Matrice réduite.

∞	60	0	10	90	20	30
70	∞	60	40	0	80	20
0	50	∞	10	60	50	30
0	20	0	∞	30	10	40
100	0	40	50	∞	50	20
0	50	30	0	20	∞	50

$$30 + 20 + 30 + 40 + 20 + 50 = 190$$

Matrice réduite en colonne.

∞	60	0	10	90	10
70	∞	60	40	0	70
0	50	∞	10	60	40
0	20	0	∞	30	0
100	0	40	50	∞	40
0	50	30	0	20	∞
0	0	0	0	0	10

10

Matrice réduite en colonne.

∞	60	0	10	90	10
70	∞	60	40	0	70
0	50	∞	10	60	40
0	20	0	∞	30	0
100	0	40	50	∞	40
0	50	30	0	20	∞
0	0	0	0	0	10

10

Au total : $190 + 10 = 200$

Calcul du regret

Dans toutes les cases où il y a un 0: $\min(\text{ligne}, \text{colonne})$.

- On privilégie le parcours de regrêr maximal.
- On étiquette alors le sommet (i, j) par le min total plus le regrêr.
- On met à jour la matrice en éliminant la ligne et la colonne concernées.

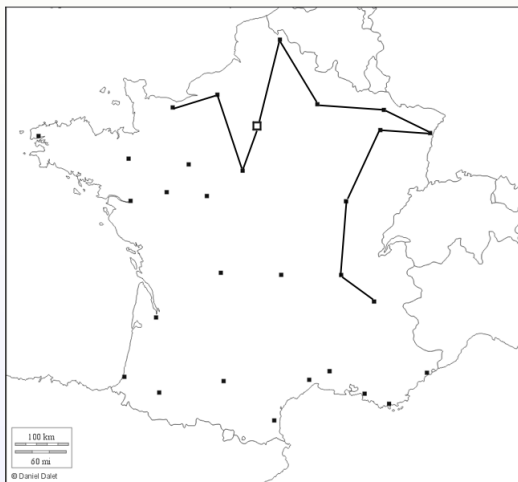
Une autre méthode de B-and-B

On cherche une borne inférieure.

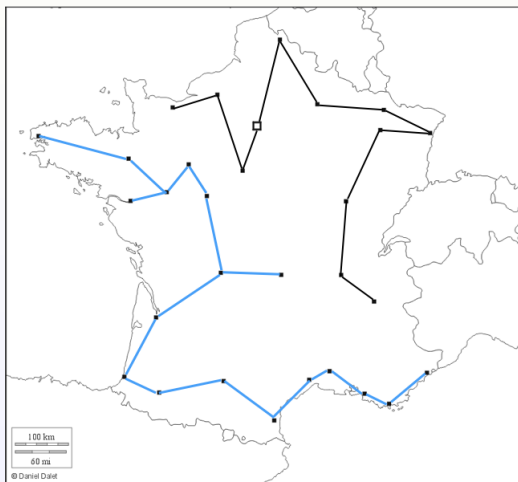
- Un arbre couvrant de poids min est calculable rapidement.
- Un chemin hamiltonien est un arbre particulier.

Comment relier cela au problème ?

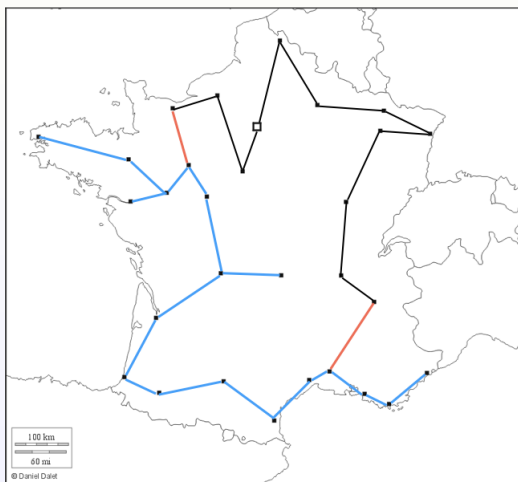
En image : solution partielle



arbre couvrant sur les villes non visitées



on relie cette solution au parcours partiel



2-opt

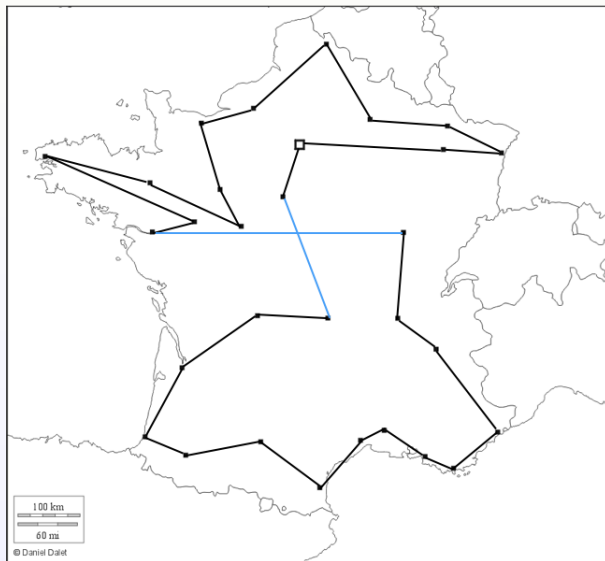
L'idée de cette heuristique est de partir d'une solution et de l'améliorer².

- On tire (au hasard) un couple d'arêtes
- On échange
- On garde la solution si elle améliore l'objectif global

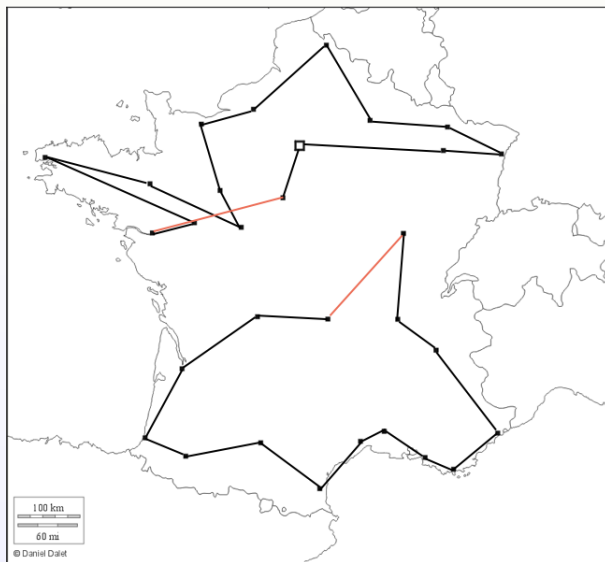
2-opt est efficace en pratique, mais n'a aucune garantie de performance vis-à-vis de l'optimum.

²on essaye de décroiser les arêtes

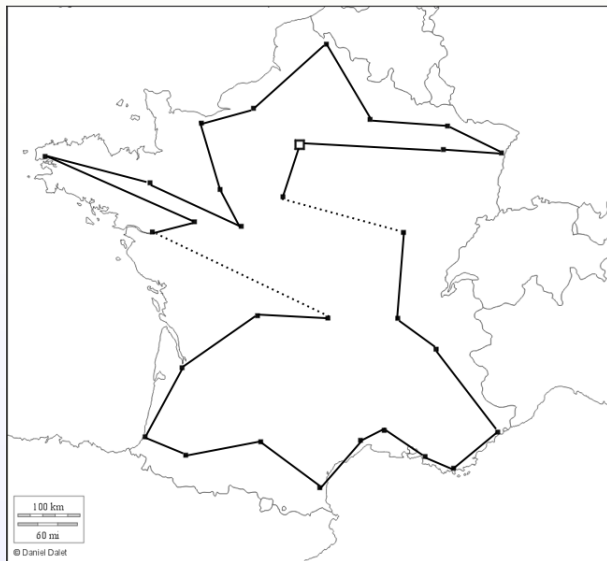
2-opt en images



2-opt mouvement interdit



2-opt



Métrique euclidienne

- L'hypothèse est l'ingalité *triangulaire*.

Interprétation : la distance directe entre deux villes est toujours plus courte que passer par une ville intermédiaire.

It is a discretization of the geometric postulate, *A straight line is the shortest distance between two points.*

TSP euclidien avec approximation garantie

Algorithme de Christofides.

Proposition

There exists an algorithm for the Euclidean TSP that produces tours whose costs are no greater than $3/2$ times the cost of an optimal tour.

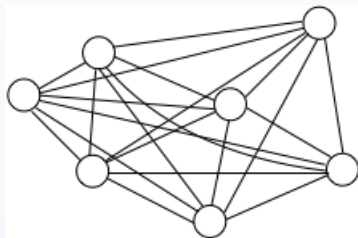
Nous allons prouver un résultat un peu plus faible, mais aussi plus simple avec 2-approximation.

Une instance de TSP euclidien

avec $n = 7$ villes.

City	Inter-City Costs						
C_1	0	$c_{1,2}$	$c_{1,3}$	$c_{1,4}$	$c_{1,5}$	$c_{1,6}$	$c_{1,7}$
C_2	$c_{2,1}$	0	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$	$c_{2,7}$
C_3	$c_{3,1}$	$c_{3,2}$	0	$c_{3,4}$	$c_{3,5}$	$c_{3,6}$	$c_{3,7}$
C_4	$c_{4,1}$	$c_{4,2}$	$c_{4,3}$	0	$c_{4,5}$	$c_{4,6}$	$c_{4,7}$
C_5	$c_{5,1}$	$c_{5,2}$	$c_{5,3}$	$c_{5,4}$	0	$c_{5,6}$	$c_{5,7}$
C_6	$c_{6,1}$	$c_{6,2}$	$c_{6,3}$	$c_{6,4}$	$c_{6,5}$	0	$c_{6,7}$
C_7	$c_{7,1}$	$c_{7,2}$	$c_{7,3}$	$c_{7,4}$	$c_{7,5}$	$c_{7,6}$	0

Représentation graphique



On note ω_G le *poids* total du graphe G (i.e. la somme des poids de ces arêtes).

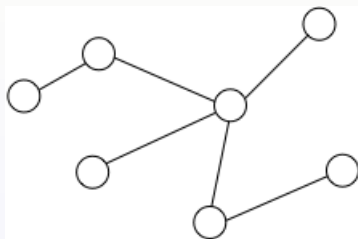
L'algorithme de Christofides procède en trois étapes qui sont tour tour étudiées dans les slides suivants.

Step 1.

Determine a minimal weight spanning tree T^* .

- A minimal weight spanning tree can be determined in polynomial time.
- ω_{T^*} est une borne inférieure de la valeur du tour optimal ω_{H^*} .

Indeed, H^* is a cycle, then, removing any edge in H^* leads to a chain, which is a particular spanning tree. As T^* is the minimal spanning tree, we have: $\omega_{T^*} \leq \omega_{H^*}$.

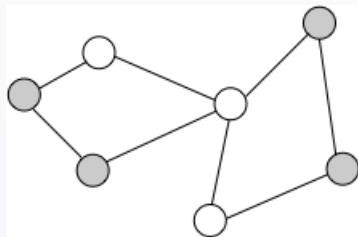


Step 2.

Consider now the set V_{odd} of the vertices of T^* whose degrees are odd.

Recall the proof that the cardinality of V_{odd} is even.
It was one in my last lecture.

Construct the perfect matching C^* of minimum weight between the vertices in V_{odd} .

T^* plus C^* 

Determine a lower bound of the optimal tour H^* .

Proof

ω_{C^*} is a lower bound of the value of the optimal tour
($\omega_{C^*} \leq \omega_{H^*}$).

Indeed, consider first the perfect matching C^* . As its vertices belong to H^* , ω_{C^*} is lower than the piece of Hamiltonian tour contained between these vertices because of the euclidian property (see Figure).

Similarly for the *complementary* perfect matching C see the second Figure.

Thus, the weight of the cycle formed by the concatenation of both perfect matchings is lower than the Hamiltonian tour

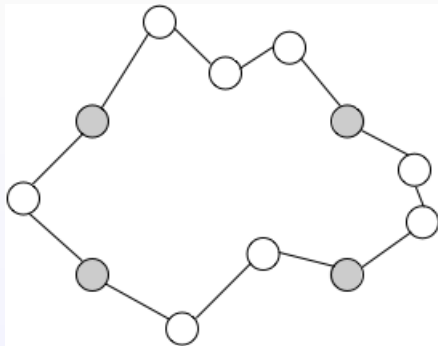
$$\omega_{C^* \cup C} \leq \omega_{H^*}.$$

Moreover, as C^* is the minimum perfect matching, we have

$\omega_{C^*} \leq \omega_C$, this concludes the proof.

Perfect Matching 1

Here is an insight of how does it works...



Perfect Matching 2

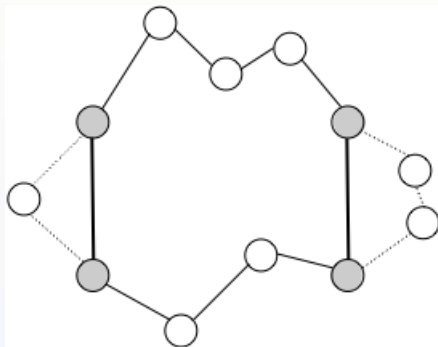


Figure: Perfect matching C^* between the vertices of odd degrees.

Step 3.

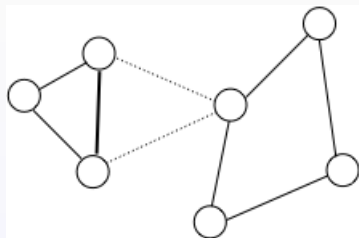
Show why all the vertices of $T^* \cup C^*$ have an even degree

We are now going to transform this graph by replacing iteratively the high degree vertices by shortcuts, which decreases the degree until reaching 2.

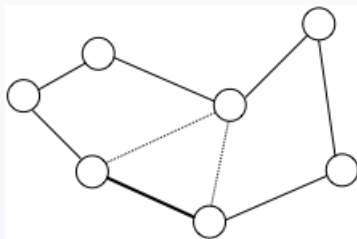
While it exists a vertex of degree greater than 4, we remove two of these consecutive edges and replace them by the opposite edge of this triangle without disconnecting the graph.

There are 2 ways to remove 2 edges and replace them by the triangle edge. Some of them disconnect the graph and thus, must be avoided.

Disconnected solution



Connected solution



Final touch

This process leads to a feasible tour.

Show that such transformations do not increase the total weight.

Finalement comme $\omega_T^* \leq \omega_H^*$ and $\omega_C^* \leq \omega_H^*$,

On en déduit que la valeur d'un tel tour est plus petite que $2\omega_H^*$.