TSP Euclidien (Travel Salesman Problem)

octobre 2016
Denis TRYSTRAM

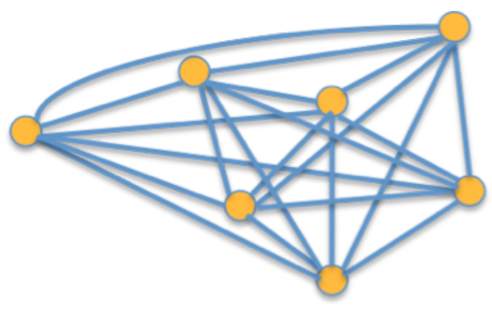
1. Définition de TSP

Informellement, le problème TSP est le suivant :
« A salesman who wants to organize the visit of his clients as best as possible. It consists in visiting them in various cities with his vehicle. Of course, he-she must go in every cities and his objective is to minimize the total distance done with his vehicle. The only information he has is the list of the cities and a map with all inter-cities distances. »
On se restreint ici au cas où les distances sont euclidiennes. C'est-à-dire :

TSP
Instance : n « villes », notées v1 jusqu'à vn et la distance entre les villes i et j est notée di,j.
Question : determiner un tour de longueur minimale qui visite chaque ville une fois et une seule.
On note OPT la valeur d'une tournée optimale.

2. Modélisation

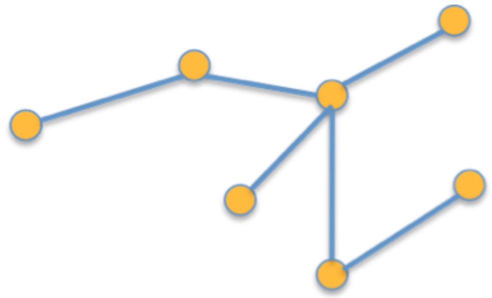
Ce problème se modélise facilement à partir des graphes. Nous allons construire un algorithme "efficace" pour résoudre ce problème. Il est bien connu qu'il est "difficile" (NP-difficile), ce qui signifie que l'on ne peut espérer construire une solution exacte en temps polynomial.
Le problème peut être formalisé comme « trouver un cycle hamiltonien de longueur minimale dans un graphe complet valué ».



3. Algorithme de Christophides

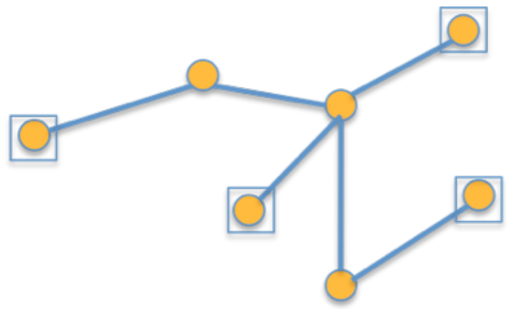
On propose ici de présenter l'analyse d'un algorithme efficace qui permet de résoudre ce problème. Il procède en trois phases décrites ci-dessous.

Phase 1. Determiner un arbre couvrant de poids minimum (c'est un algorithme connu, linéaire en nombre d'arêtes – après un tri en ordre croissant). On le note T*. D'une manière générale, on notera ω_G le poids total du graphe G (somme des poids des arêtes).

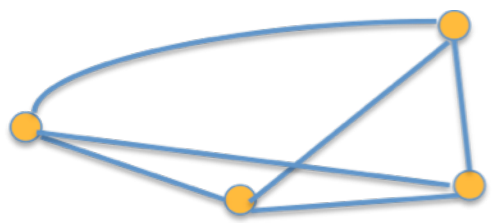


En particulier, ω_{T^*} est une borne inférieure de la valeur du tour optimal. En effet, si l'on retire une arête dans un tour optimal (qui est un cycle hamiltonien) on obtient une couverture du graphe. On note ω_{T^*} la valeur minimale du poids d'un arbre couvrant.

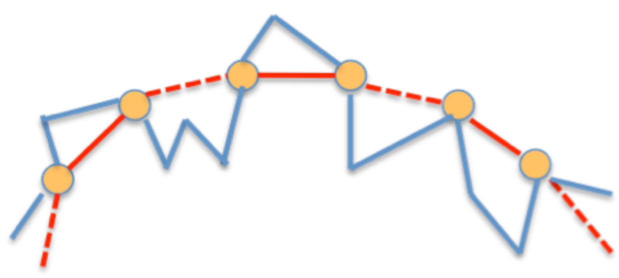
Phase 2. Considérons maintenant l'ensemble V_O des sommets de T* dont les degrés sont impairs. Il est facile de démontrer que la cardinalité de V_O est paire (c'est un résultat classique en Théorie des graphes...). Sur l'exemple, il y a 4 sommets.
On construit maintenant un couplage parfait C* de poids minimum entre ces sommets de V_O.
On montre maintenant que $2\omega_{C^*}$ est une borne inférieure du tour optimal.



C* est un couplage parfait du graphe suivant réduit à V_O :

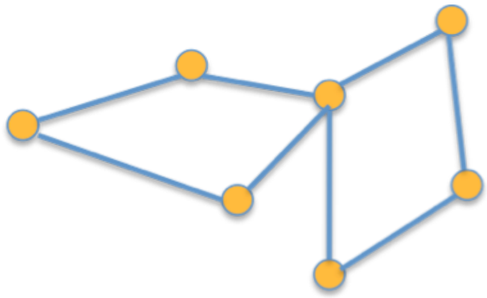


Considérons un tour optimal. Il contient en particulier les sommets de V_O. Supposons qu'ils soient ordonnés suivant leur rang dans ce tour. Comme le montre la figure suivante, on considère en plus de C* un autre couplage M entre les sommets de degrés impairs (C* est en traits pleins en rouge, M2 en hachuré, le tour optimal est en bleu).



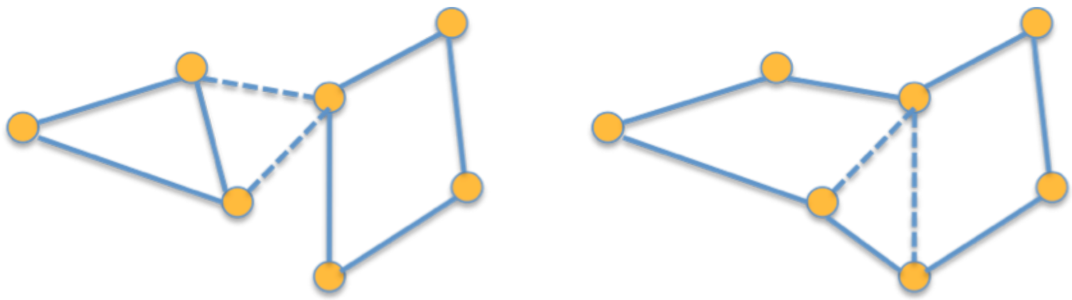
En utilisant l'inégalité triangulaire, on déduit :
$OPT \geq \omega_{C^*} + \omega_M$
Cependant, C* est un couplage minimum, donc, ω_{C^*} est une borne inférieure de tout autres couplage, en particulier M, ainsi : $\omega_M \geq \omega_{C^*}$
donc : $OPT \geq 2\omega_{C^*}$
On the example, the graph T* ∪ C* is the following :

Phase 3. On part du graphe G=T* ∪ C* et on le modifie pour obtenir une tournée. Pour notre exemple, ce graphe est le suivant :



All the vertices of the graph T* ∪ C* have an even degree because the vertices of T* of even degree are not changed and the vertices with odd degree have one extra edge, leading to an even degree.

The algorithm proceeds iteratively by removing two of the consecutive edges of the vertices whose degree is strictly greater than 2 and replacing them by the opposite edge of this triangle without disconnecting the graph (as it is shown in the figure below, there are several ways to do this, the first solution disconnects the graph and leads to a not feasible solution).



We are going to transform this graph in the following way: We will replace iteratively some edges by shortcuts. 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.

The previous process leads to a feasible tour by construction because there exists an edge at each iteration that does not disconnect the graph where the degree of the considered vertex decreases by 2 (as illustrated in the figure, from 4 to 2).

A each iteration, such transformations do not increase the total weight because of the triangular inequality : the added edge has a weight lower than both other sides of the triangle.

Finally, we can conclude by computing the approximation ratio of this algorithm :
We established in question 2 that $\omega_{T^*} \leq OPT$, then, in question 4 that $2\omega_{C^*} \leq OPT$. In question 6, we prove that the solution is feasible, and from the last question that the total weight never increases. Thus,

$\omega \leq \omega_{T^*} + \omega_{C^*} \leq OPT + \frac{1}{2} OPT$
Cet algorithme est donc une 3/2-approximation