

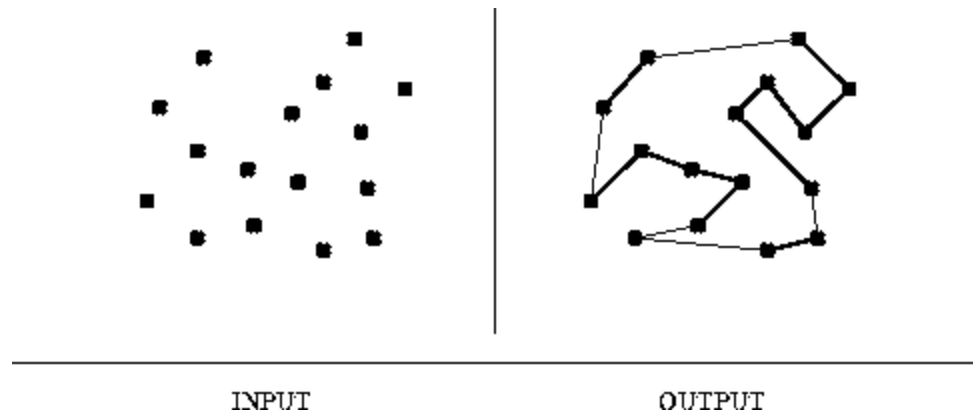
Voyageur de commerce

Résolution pratique - M6601 Recherches Opérationnelles

M. Vanlerberghe - L. LeBohec - N. Masson

Introduction

Le **problème du voyageur de commerce** (TSP), est un problème d'optimisation qui détermine un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ.



Malgré la simplicité de son énoncé, il s'agit d'un problème d'optimisation pour lequel on ne connaît pas d'algorithme permettant de trouver une solution exacte rapidement dans tous les cas. Plus précisément, on ne connaît pas d'algorithme en **temps polynomial** qui permettrait de le résoudre.

Introduction	1
Démarche & Stratégies	2
Difficultés	3
Résultat des tests	4
Distance	4
Résultats avec le recuit:	4
Comparaison avec 20 villes:	5
Conclusion	6
Sources	7

Démarche & Stratégies

Afin de résoudre le problème du voyageur de commerce, nous nous sommes concentrés sur trois algorithmes:

- Heuristique par ajout du plus proche
- Descente locale
- Recuit simulé (Simulated Annealing)

Nous avons essayé d'utiliser l'heuristique comme parcours de départ pour la descente locale et le recuit simulé (voir comparatif).

L'entrée dans le programme se fait via des fichiers de données. Les variables du problème sont donc:

- Le nombre de villes
- Les coordonnées de chaque ville

Pour le recuit, deux variables supplémentaires sont nécessaires:

- La température initiale
- Le refroidissement du système

Dans nos jeux de tests, ces variables ont été définies de façon empirique.

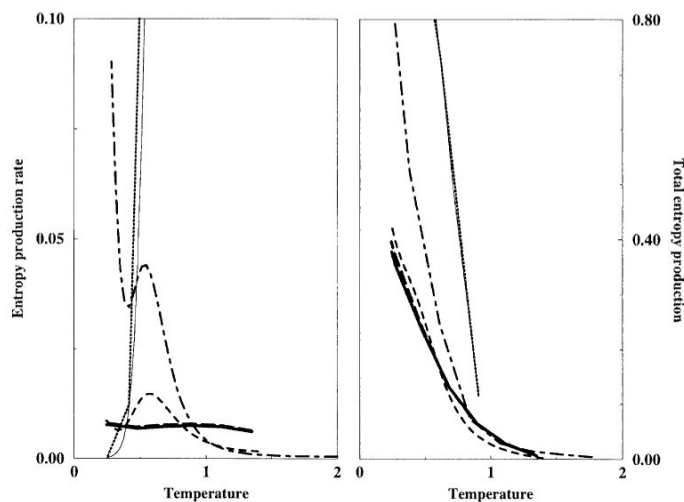
Difficultés

Pour la descente locale:

- J'avais du mal à comprendre la notion de parcours voisin, savoir quels points il fallait permuter entre eux.

Pour le recuit:

- Les principaux inconvénients du recuit simulé résident dans le choix des nombreux paramètres, tels que la **température initiale**, la loi de **décroissance de la température**, les critères d'arrêt ou la longueur des paliers de température. Ces paramètres sont souvent choisis de manière **empirique**.
- Il était difficile de trouver une température initiale et un refroidissement correct, afin d'optimiser les résultats et le temps d'exécution de notre algorithme.



Afin de surmonter cette difficulté, nous avons consulté l'article **A comparison of simulated annealing cooling strategies** (lien disponible dans la partie "sources").

Nous avons tiré de cet article qu'il valait mieux choisir un refroidissement très bas, et que la température baisse de façon exponentielle si elle est choisie très haute.

Résultat des tests

Distance

	20 villes	100 villes	1000 villes
Heuristique	105.52	8431.85	128465.80
Descente locale à partir du jeu d'essai	142.27	41451.88	3054618.4
Descente locale à partir de la solution de l'heuristique	100.68	8380.52	128182.1
Recuit simulé à partir d'un parcours aléatoire	91.94	8417.96	466320.916
Recuit simulé à partir de l'heuristique	91.57	8355.33	128465.80

*recuit simulé avec une température de 1000 et un refroidissement de 0.00001

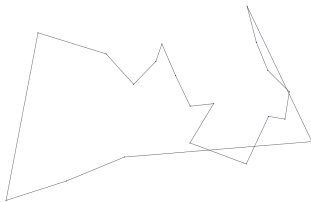
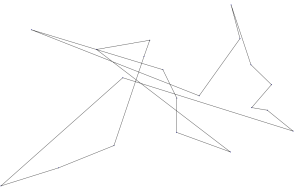
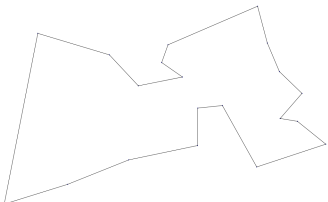
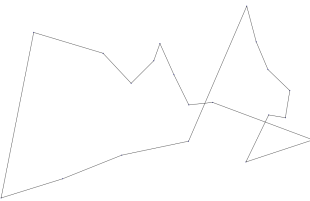
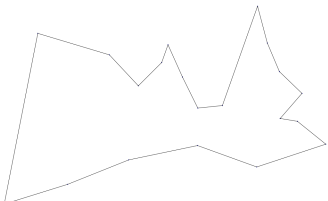
Résultats avec le recuit:

Pour ces résultats, nous avons utilisé la même température et le même refroidissement pour tous les jeux d'essai.

Il serait judicieux de changer ces paramètres afin de trouver de meilleurs résultats.

Puisque le chemin initial du recuit est pris au hasard au début de l'algorithme, il reste une part de hasard dans le résultat qui le rend différent d'une exécution sur l'autre.

Comparaison avec 20 villes:

	Heuristique	Descente locale	Recuit
A partir du jeu d'essai			
Raffinée par heuristique			

Conclusion

La solution optimale trouvée est satisfaisante.

Nous aurions toutefois aimé avoir le temps d'implanter des solutions supplémentaires, par exemple:

- Le **raffinement du recuit** par lui même. En effet, nous aurions pu effectuer plusieurs cycles de recuit simulé les uns à la suite des autres afin d'obtenir un meilleur résultats.
- Nous aurions pu également expérimenter la **méthode tabou**, et la comparer aux autres méthodes. Par manque de temps, nous ne l'avons pas fait.

Nous avons observé que **le recuit ne se suffit pas à lui même**. En effet, puisqu'au début de l'algorithme, il prend un cycle aléatoire, les itérations échangeant une à une les villes ne sont pas suffisantes lorsque le nombre de villes devient plus grand.

Il est donc intéressant de coupler au recuit une solution permettant de donner un résultat s'approchant de celui désiré, puis de raffiner la solution grâce au recuit.

Sources

- **A comparison of simulated annealing cooling strategies** , Université de Copenhague, 1997.
<https://www.fys.ku.dk/~andresen/BAhome/ownpapers/permanents/annealSched.pdf>
- **Travelling Salesman Problem** sur Wikipédia
https://en.wikipedia.org/wiki/Travelling_salesman_problem