

Méthodes Computationnelles

Projet

Problème du voyageur de commerce

Analyse du problème

Le problème du voyageur de commerce nous met face à, comme son nom l'indique, un voyageur de commerce, qui doit visiter plusieurs villes pour affaire, et qui souhaite optimiser son trajet.

Le problème peut avoir plusieurs paramètres, tel que le nombre de villes à visiter, et leurs position. L'objectif est de minimiser le coût (ici la distance parcourue) de son voyage.

Solutions mises en œuvre

Un algorithme déterministe pourrait résoudre ce problème, mais sa complexité $O(n!)$, avec n étant le nombre de villes. Cette complexité rend inenvisageable le calcul rapide d'un trajet avec un nombre élevé de villes.

Le choix d'un algorithme d'approximation s'impose donc, contrairement au déterministe, il ne garantit pas de trouver la meilleure solution, mais il permet de trouver l'une des meilleures, pour une complexité bien moindre. Dans notre cas, l'adoption d'une solution approchée de l'optimale est envisageable.

Nous allons ici utiliser un système immunitaire artificiel (SIA, ou AIS en anglais), utilisant un algorithme par sélection clonale.

Cette algorithme fonctionne d'une manière simple :

- Génération de N anti-corps avec une solution,
- Sélection des meilleurs anti-corps (ceux au coût le moins élevé),
- Clonage des meilleurs anti-corps,
- Mutation des clones,
- Sélection des meilleurs anti-corps et clones mutés confondus,
- Mutation des moins bons / Remplacement des moins bons,
- Recommencement / Fin.

Nous pouvons ainsi faire varier plusieurs paramètres :

- Le nombre de générations (de cycle ci-dessus),
- Le nombre d'anti-corps,
- Le pourcentage des meilleurs anti-corps à cloner,
- Le pourcentage des pire anti-corps à remplacer,

- Le fréquence de remplacement des pires anti-corps,
- La méthode de mutation (mutation par échange de deux villes, par inversion des villes entre deux, par translation des villes entre deux vers une autre position),
- La méthode de sélection (sélection du meilleur entre le clone et sont anti-corps d'origine, sélection des meilleurs anti-corps et clones confondus),
- Le nombre (maximum) de mutation (nombre fixe, variable en fonction de la position de l'anti-corps au classement des coûts).

Nous cherchons donc à obtenir la combinaison de ces paramètres donnant un coût optimal.

Pour notre procédure de recherche, nous allons essayer faire jouer les différents paramètre sur 2 cartes, contenant 30 et 100 villes.

Méthodologie

Afin d'avoir un nombre de tentative suffisante pour établir des pourcentage et calculer des moyennes, nous avons écrit un script bash qui effectue les calculs de coût 20 fois, en établissant une moyenne, modifie certains paramètres et recommence avec ces nouveaux paramètres. Afin d'automatiser la tâche, nous avons ajouté une directive au compilateur inhibant l'affichage de plots et les détails des résultats pour ne renvoyer uniquement le meilleur coût ce qui explique le manque de graphique dans ce document.

Analyse des stratégies et de l'influence des paramètres

Tous les résultats de coût de l'algorithme indiqués ci après sont issues d'une moyenne de 20 tentatives avec les mêmes paramètres.

L'expression « autres paramètres variés » signifie que les paramètres autres que ceux déjà indiqués sont attribués de manière à représenter un large panel de possibilités. Ces autres paramètres restent les mêmes entre chaque test interne à un paramètre.

Nombre d'anti-corps (population):

Tests avec des tailles de population de 100, 350, et 1000, et les autres paramètres variés :

- 1000 : coût de référence (100%)
- 350 : 4,29% de coût en plus que 1000
- 100 : 6,75% de coût en plus que 1000

Comme on pouvait l'imaginer, plus le nombre d'anticorps est élevé, plus les coûts obtenues sont bas.

Tests suivants effectués avec 350 anticorps, nombre moyen, pour garder des résultats imparfaits et observer des différences dans les tests suivants.

Nombre de génération :

Tests avec des tailles de population de 100, 350, et 1000, et les autres paramètres variés :

- 1000 : coût de référence (100%)
- 350 : 2,63% de coût en plus que 1000
- 100 : 25,33% de coût en plus que 1000

Comme on pouvait l'imaginer, plus le nombre de génération est élevé, plus les coûts obtenues sont bas.

Tests suivants effectués avec 350 générations, nombre moyen, pour garder des résultats imparfait et observer des différences dans les tests suivants.

Nombre de clones :

Tests avec les pourcentages de clones par rapport à la population (350) : 25, 50, et 75, et les autres paramètres variés :

- 75%: coût de référence (100%)
- 50%: 0,02% de coût en plus que 75%
- 25%: 4,29% de coût en plus que 75%

Plus le nombre d'AC est élevé, plus la coût et réduit : Cela s'explique par le fait que le nombre de clone grossit artificiellement la population, entraînant un plus fort nombre de mutations.

Tests suivants effectués avec 45% de clones, pour garder des résultats imparfait et observer des différences dans les tests suivants.

Nombre et fréquence de nouveaux anti-corps :

Tests avec des fréquences d'injection de 25, 85, 175 (pour un nombre de générations de 350), et les pourcentages de pires remplacés par des nouveaux AC par rapport à la population (350) : 25, 50, et 75, et les autres paramètres variés :

- 25 :
 - 25%: 0,12% de coût en plus que 75%
 - 50%: 0,77% de coût en plus que 75%
 - 75%: coût de référence (100%)
- 85 :
 - 25%: 0,96% de coût en plus que 75%
 - 50%: 0,99% de coût en plus que 75%
 - 75%: coût de référence (100%)
- 175 :
 - 25%: coût de référence (100%)
 - 50%: 1,16% de coût en plus que 25%
 - 75%: 0,90% de coût en plus que 25%

Pour ces paramètres, il y a deux tendances qui se présentent : soit on remplace rarement peu d'AC par des nouveaux, soit on remplace fréquemment beaucoup d'AC par des nouveaux.

Cette première tendance permet aux moins bons AC d'avoir le temps de beaucoup mutés, pour trouver une meilleure solution.

Cette seconde tendance permet aux moins bons AC d'être remplacé fréquemment, et donc d'augmenter la probabilité de trouver une solution intéressante par hasard.

Tests suivants effectués avec les 9 combinaisons de valeurs présentées ci-dessus.

Méthode de sélection

Tests avec les différentes méthode de sélection : la sélection 2 à 2, et la sélection par tri de sous-populations, ainsi que les autres paramètres variés :

- tri de sous-population : coût de référence (100%)
- 2 à 2 : 25,16% de coût en plus que celle par tri de sous-populations

La méthode de sélection en tri de sous-populations est nettement plus efficace que celle en 2 à 2, car on tri globalement, ce qui fait que, si un clone est son AC d'origine sont tout les deux les plus efficaces, on pourra tout les 2 les sélectionner, contrairement à la méthode en 2 à 2.

Méthode de mutation

Tests avec les différentes méthode de mutations : la mutation par échange, la mutation par inversion, et la mutation par translation, ainsi que les autres paramètres variés :

- échange : 11,35% de coût en plus que la mutation par inversion
- inversion : coût de référence (100%)
- translation : 19,13% de coût en plus que la mutation par inversion

La méthode de mutation par inversion est nettement plus efficace que les deux autres, car elle permet de conserver une suite de visites, contrairement à celle par échange, elle permet aussi de ne pas briser une suite, contrairement à celle par translation.

Nombre de mutations

Tests avec différents nombre de mutations : un nombre de mutations statique (1,3,10), et un nombre de mutations dynamique (max : 3,10) (dynamique : le nombre de mutations dépend de la position de l'AC dans le classement par coût), ainsi que les autres paramètres variés :

- statique: Pour 1 mutation, coût de référence (100%)
 - 1 : coût de référence (100%)
 - 3 : 1,59% de coût en plus qu'avec 1 mutation en statique
 - 10 : 3,73% de coût en plus qu'avec 1 mutation en statique
- dynamique: 1,45% de coût en plus pour 3 mutations maximum en dynamique que pour 1 mutation en statique
 - 3 : coût de référence (100%)
 - 10 : 2,55% de coût en plus qu'avec 10 mutations en dynamique

Le meilleur nombre de mutation s'avère être le plus faible (1), donc en statique (en dynamique avec 1 mutation = statique avec 1 mutation), cela s'explique par le fait qu'un nombre élevé de mutation revient à trop s'éloigner de l'AC d'origine, et donc à « créer » un nouvel AC.

Conflits

Le nombre de clone et nombre de renouvellement dépendent de la population. Si on met les deux valeurs à 100%, l'intégralité des AC seront renouvelés avant d'être supprimés, ce qui devient contre-productif, il faut donc trouver des valeurs telles que les clones et les nouveaux ne se traitent pas les mêmes AC.

Conclusion

On peut déduire de cette analyse que les combinaisons idéales de paramètres sont de soit maximiser le nombre d'AC à faire muter, soit de maximiser le nombre d'AC neuf qu'on génère. Un nombre élevé de génération et une population de base élevée tendent vers un bon résultat.

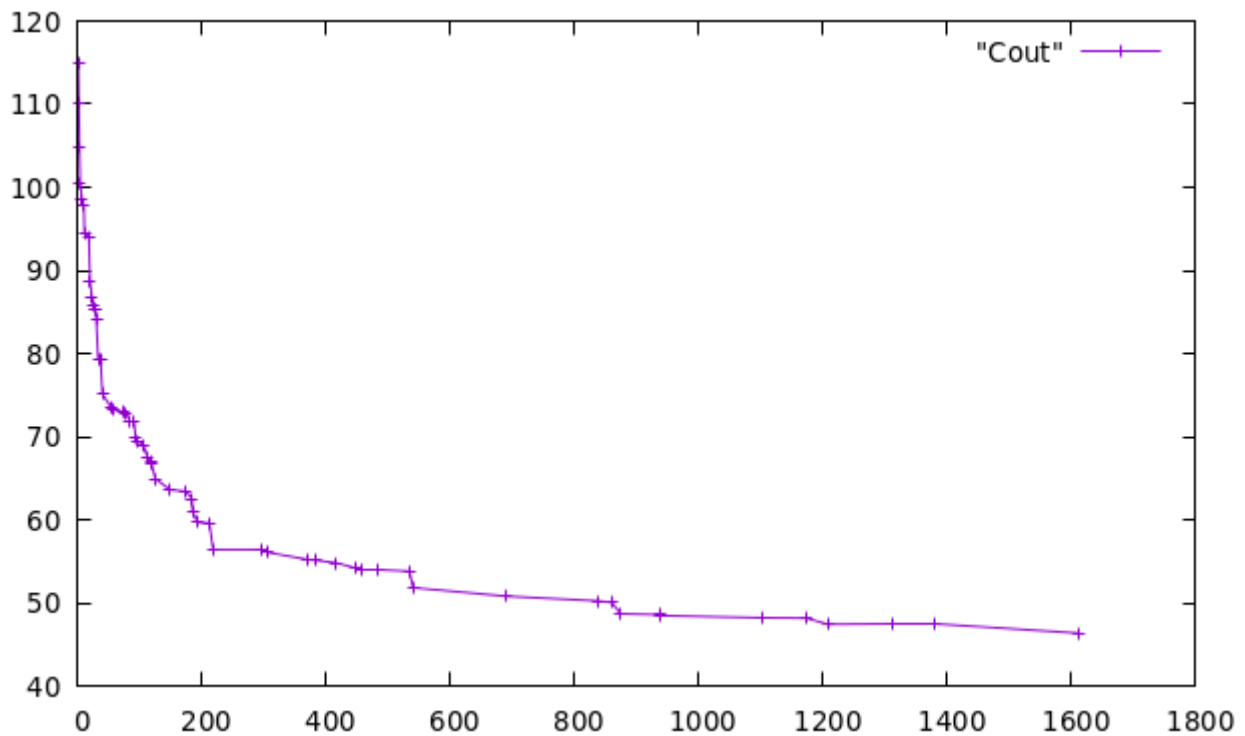


Figure 1: Meilleur résultat possible à 30 villes

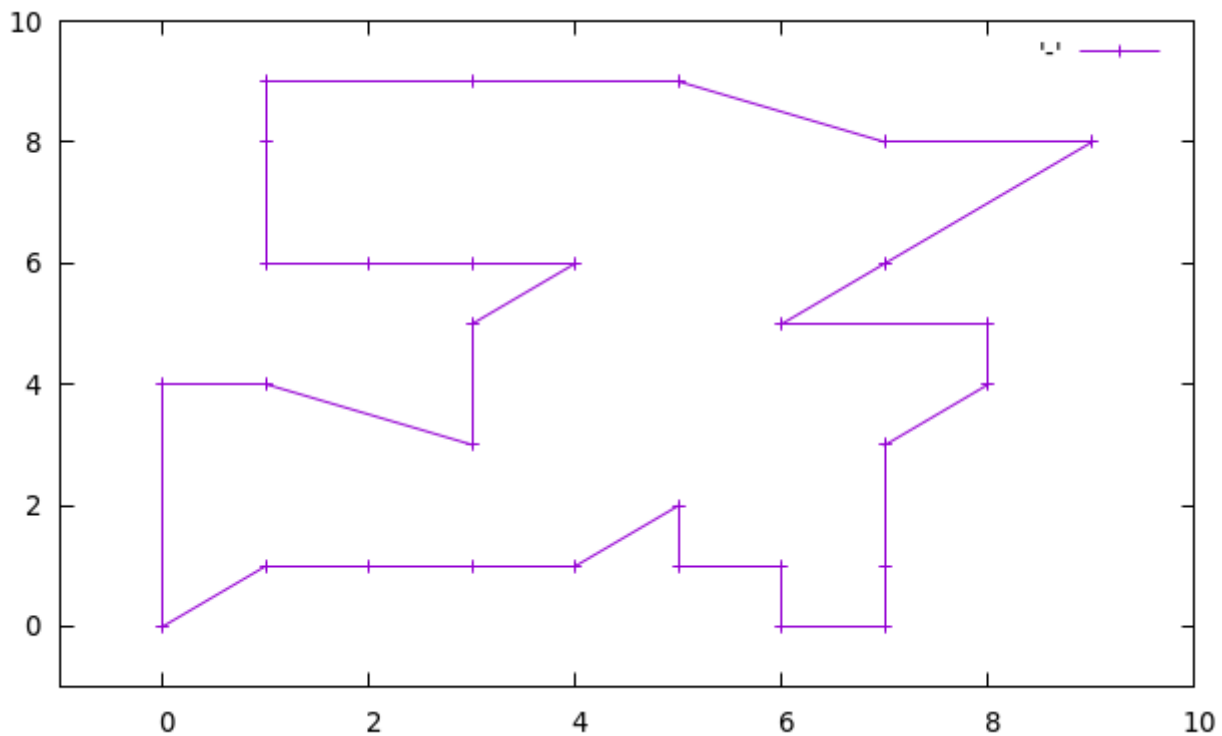


Figure 2: Chemin le plus optimisé

Les deux figures précédentes présentent le résultat le plus optimisé pour 30 villes. Les meilleurs paramètres sont les suivants
3500 individus, 45 % de clone, 25 % de nouveau, 1650 génération et nous injectons des nouveaux toutes les 75 générations. Nous avons utilisé la méthode de mutation de la translation en utilisant un tri de sous-populations.

Notons que l'optimisation de ses paramètres aurait pu être faite à l'aide d'un système immunitaire artificiel, cherchant une solution approchée de l'optimale.