Résolution du Problème du Voyageur de Commerce (TSP) à l'aide de l'Algorithme Multi-Level Cross Entropy (MCEO)

Rhimini Aimane
Supervisé par : Pr. Khalid Jebbari
Faculté des sciences et techniques de Tanger (FSTT)
24 juin 2025

Implémentation disponible sur GitHub

Résumé

Ce travail explore une méthode innovante pour résoudre le Problème du Voyageur de Commerce (TSP) en utilisant l'algorithme évolutif Multi-Level Cross Entropy Optimizer (MCEO).

Contrairement aux approches classiques, le MCEO combine robustesse et efficacité, ce qui le rend particulièrement adapté aux problèmes d'optimisation combinatoire à grande échelle.

L'étude se base sur un ensemble réel de villes marocaines, avec une modélisation précise des distances via la formule de Haversine.

L'objectif principal est de démontrer la flexibilité et la scalabilité du MCEO, ouvrant des perspectives prometteuses pour des applications en logistique et planification urbaine.

Introduction

L'optimisation combinatoire représente un domaine essentiel de la recherche opérationnelle, avec des applications concrètes dans la logistique, l'ingénierie, les télécommunications, et bien d'autres secteurs.

Les problèmes de ce type sont souvent complexes, notamment lorsqu'ils sont classés comme NP-difficiles, ce qui signifie qu'aucun algorithme connu ne peut les résoudre efficacement de manière exacte à grande échelle.

Pour pallier cette difficulté, les méthodes dites "méta-heuristiques" ont été développées. Elles sont capables d'explorer intelligemment des espaces de recherche immenses afin d'identifier des solutions de bonne qualité, sans exiger une connaissance approfondie de la structure du problème. Parmi ces approches, l'algorithme Multi-Level Cross Entropy Optimizer (MCEO).

1. Le Problème du Voyageur de Commerce

Le TSP est un problème d'optimisation combinatoire dans lequel un agent ou un voyageur doit parcourir un ensemble de n villes en minimisant la distance totale parcourue, tout en visitant chaque ville une seule fois et en revenant à son point de départ.

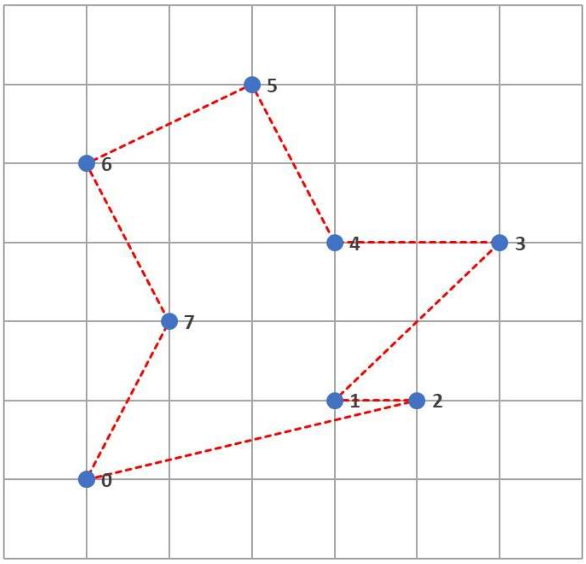

Figure 1: Parcours optimal représentant la solution du TSP

Mathématiquement, il s'agit de trouver la permutation optimale π des villes qui minimise la fonction objective suivante :

$$C(\pi) = \sum_{i=1}^{n} d(\pi_i, \pi_{i+1})$$

Avec $\pi_{n+1} = \pi_1$ pour assurer le retour au point de départ, et $d(i, j)$ la distance entre les villes i et j.

La complexité combinatoire du TSP est telle que le nombre total de solutions possibles est de pour n villes.

Applications du TSP

Le TSP est largement étudié et utilisé dans :

- La logistique (itinéraires de livraison).
- L'électronique (placement de composants).
- La biologie computationnelle (séquençage d'ADN).

Des algorithmes comme les colonies de fourmis, le recuit simulé ou encore les algorithmes génétiques ont été couramment utilisés pour l'approximation de sa solution.

1. Cross Entropy Optimizer

Cross Entropy Optimizer (CE) est une méthode inspirée de la théorie de l'information et largement utilisée pour la résolution de problèmes d'optimisation difficiles.

L'idée principale repose sur l'utilisation d'une distribution de probabilité pour générer des solutions aléatoires, puis sur l'ajustement progressif de cette distribution en se basant sur les meilleures solutions rencontrées.

Ce processus permet d'orienter la recherche vers des régions prometteuses de l'espace des solutions.

Le cœur de CE repose sur la minimisation de la divergence de Kullback-Leibler entre la distribution actuelle et la distribution idéale, formulée comme suit :

$$H(p, q) = -\sum_{x} p(x) \log\, q(x)$$

Malgré ses qualités, CE présente plusieurs limitations :

- **Exploration limitée** : la méthode tend à se concentrer trop sur certaines régions et négligeant d'autres.

- **Convergence prématurée** : avec un seul centre de probabilité, il est fréquent que CE se fige dans un optimum local.

- **Sensibilité aux paramètres** : en particulier la taille de l'échantillon et la sélection des élites peuvent grandement influencer les performances.

- **Scalabilité** : CE peut devenir inefficace pour les problèmes de grande dimension ou très complexes.

Ces limitations motivent le développement de variantes améliorées, comme le MCEO.

2. Multi-Level Cross Entropy Optimizer

MCEO appartient à la famille des **méta-**heuristiques évolutives, qui combinent des mécanismes d'inspiration naturelle ou probabiliste pour rechercher efficacement dans des espaces complexes et de grande dimension. Ces algorithmes ne garantissent pas une solution optimale, mais offrent un bon compromis entre qualité de solution et coût computationnel.

Multi-Level Cross Entropy Optimizer est une extension évolutive de CE qui vise à surmonter ses limitations en introduisant une structure multi-niveau améliorer sa robustesse.

Plutôt que de s'appuyer sur une unique distribution, MCEO maintient plusieurs centres de probabilité, appelés Averages, chacun capable d'explorer différentes régions de l'espace de recherche.

Chaque average correspond à une matrice de transition et est perturbé de manière stochastique à chaque itération.

Différence avec CE

La différence principale entre le Cross-Entropy Optimizer et le Multi-Level Cross-Entropy Optimizer réside dans la manière dont la déviation standard σ des Search Agents est gérée.

Contrairement à CE qui utilise une seule déviation standard décroissante au fil des itérations, MCEO ajuste dynamiquement σ pour chaque centre ou Average en fonction de sa position dans l'espace de recherche.

Cela permet une exploration adaptative multi-niveaux : plus une zone est prometteuse, plus la perturbation est fine ; moins elle l'est, plus la variance est élevée pour favoriser l'exploration.

La formule mathématique utilisée pour calculer cette déviation standard est :

$$\sigma_i = \left(\frac{i}{n_{\text{Averages}}}\right)^3 \cdot \left(\frac{H_{\text{b}} - L_{\text{b}}}{\varphi}\right), \quad \varphi = \frac{n_{\text{Var}}^2}{n_{\text{Averages}}}$$

Où :

- $\boldsymbol{\sigma_i}$ est la déviation standard pour le $\boldsymbol{i}$ - ème Average.
- $\boldsymbol{i}$ varie de 1 a $\boldsymbol{n_{averages}}$, selon le rang du centre.
- $\boldsymbol{H_b}$ et $\boldsymbol{L_b}$ sont respectivement la borne supérieure et inférieure de l'espace de recherche.
- $\boldsymbol{\varphi}$ est un facteur d'échelle tenant compte de la dimension du problème (**nVar**).

Cette formulation favorise une exploitation intensive des zones prometteuses et une exploration plus large des zones moins explorées.

Processus de Fonctionnement du MCEO

L'algorithme MCEO suit une approche itérative d'optimisation structurée autour de deux mécanismes complémentaires : l'exploration systématique de l'espace de recherche et l'exploitation ciblée des solutions prometteuses.

Algorithm – Optimisation par Cross Entropy Multi-Niveaux

Input : Ensemble des moyennes A, vecteurs de perturbation σ,
Nombre d'itérations maximal $MaxIteration$,
Facteur d'exploration α, plage SRF_{min}, SRF_{max},
Fonction de coût $Cost()$

Output : Meilleure tournée $BestTour$, coût $BestCost$

Initialiser A et σ ;
for $iteration \leftarrow 1$ **to** $MaxIteration$ **do**
  Générer des agents de chaque moyenne avec une distribution normale et σ ;
  Évaluer le coût de chaque agent avec $Cost()$;
  Trier tous les agents selon leur coût ;
  Sélectionner les meilleurs $(1-\alpha)\cdot n_{Averages}$ pour l'exploitation ;
  Remplacer les $\alpha \cdot n_{Averages}$ restants de façon aléatoire (exploration) ;
  Mettre à jour σ à l'aide du facteur SRF ;
  Mettre à jour la meilleure solution si amélioration ;
  if convergence détectée **then**
  break
return $BestTour$, $BestCost$

Figure 2: Architecture de l'algorithme MCEO

3. Approche de MCEO pour le TSP

L'adaptation de l'algorithme Multi-Level Cross Entropy Optimizer au Problème du Voyageur de Commerce repose sur une reformulation probabiliste du parcours optimal.

Chaque centre de probabilité (Average) est représenté par une matrice de transition, où chaque ligne modélise la probabilité de transition d'une ville vers une autre.

Données utilisées

L'implémentation s'appuie sur un jeu de données réel de 10 villes marocaines, modélisées par leurs coordonnées GPS, avec des distances inter-villes calculées via la formule Haversine.

Une visualisation des connexions complètes entre ces villes est présentée sous forme des points liés par des lignes, illustrant toutes les possibilités initiales.

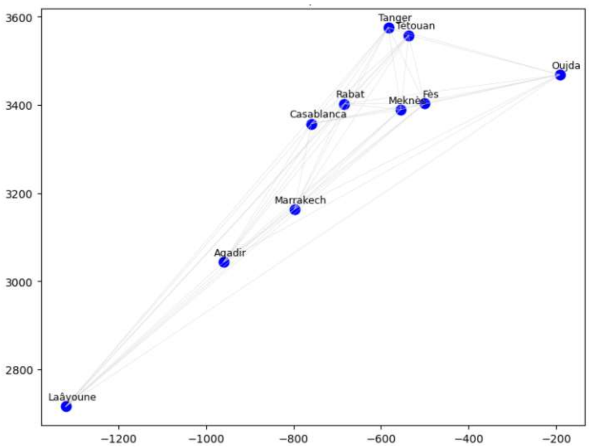

Figure 3: Graphe des connexions entre des villes marocaines

Résultats des itérations

L'algorithme MCEO a été exécuté sur 100 itérations avec 7 averages et 25 search agents par itération.

La distance diminue rapidement jusqu'à l'itération 20, puis se stabilise progressivement.

À partir de l'itération 63, la distance devient presque constante, indiquant que l'algorithme a convergé vers une solution optimale.

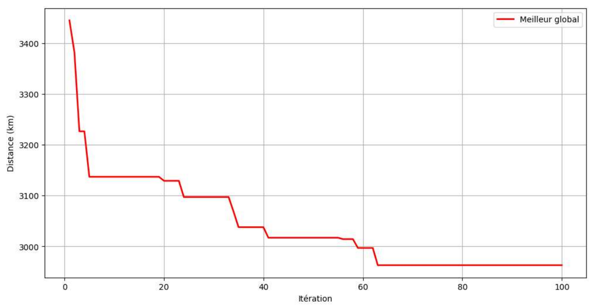

Figure 4: Graphe de convergence de l'algorithme MCEO

Les itérations ont permis de passer d'une distance initiale de 3444.77 km à une distance globale optimisée de 2963.04 km.

Itération	1	2	3	5	20	24	34	35	41	56	59	63	100
Distance	3444	3382	3226	3137	3129	3097	3068	3037	3017	3014	2996	2963	2963

Table 1: Évolution de la distance par itération

Trajet finale

Le meilleur parcours optimisé trouvé par l'algorithme est :

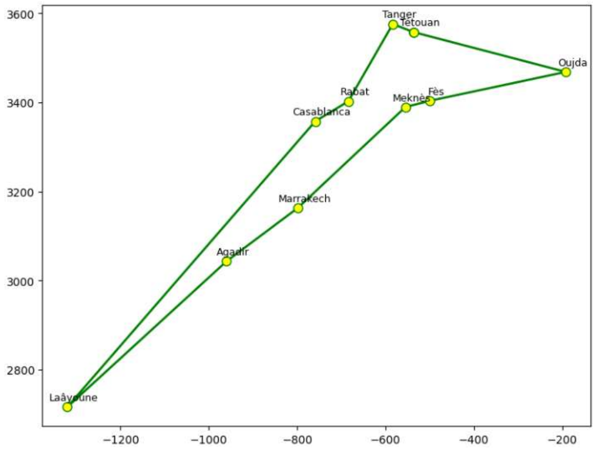

Figure 5: Tour optimal obtenu par MCEO

Conclusion

Ce projet a démontré l'efficacité de l'algorithme Multi-Level Cross Entropy Optimizer (MCEO) pour résoudre le Problème du Voyageur de Commerce (TSP) en utilisant un ensemble réel de 10 villes marocaines.

Grâce à une modélisation précise des distances via la formule de Haversine et une approche multi-niveau adaptative, le MCEO a permis de réduire la distance totale de 3444,77 km à 2963,04 km en 100 itérations, avec une convergence notable dès l'itération 63.

Cette étude met en évidence la robustesse et la scalabilité de l'algorithme, surpassant les limitations du Cross Entropy Optimizer classique, notamment en termes d'exploration et de convergence prématurée.

Les résultats ouvrent des perspectives prometteuses pour des applications pratiques en logistique et planification urbaine, bien que des ajustements supplémentaires, tels qu'une augmentation du nombre d'itérations ou d'agents de recherche, pourraient encore améliorer la précision.

Cette recherche constitue une base solide pour de futures explorations, notamment avec des ensembles de données plus vastes ou des contraintes supplémentaires.

Références

[1]. Kaveh, A., & Bakhshpoori, T. (2017). Multi-level cross entropy optimizer (MCEO): an evolutionary optimization algorithm for engineering problems. Engineering with Computers, 34(2), 309-326. https://link.springer.com/article/10.1007/s00366-017-0569-z

[2]. Engineering with Computers. (n.d.). https://link.springer.com/journal/366

[3]. Botev, Z. I., Kroese, D. P., & L'Ecuyer, P. (n.d.). The Cross-Entropy Method for Optimization. Université de Montréal & The University of Queensland.

[4]. V7Labs. (n.d.). Cross Entropy Loss: Intro, Applications, Code. Retrieved from https://www.v7labs.com/blog/cross-entropy-loss-guide

[5]. Ivy Mobility Developers. (n.d.). Traveling Salesman Problem. Retrieved from https://medium.com/ivymobility-developers/traveling-salesman-problem-9ab623c88fab

[6]. BaseCS. (n.d.). The Trials And Tribulations Of The Traveling Salesman. Retrieved from https://medium.com/basecs/the-trials-and-tribulations-of-the-traveling-salesman-56048d6709d

[7]. Kroese, D. P., & Rubinstein, R. Y. (2006). The Cross-Entropy Method for Continuous Multi-Extremal Optimization. *Methodology and Computing in Applied Probability*, 8(3), 383-407. https://link.springer.com/article/10.1007/s11009-006-9753-0