

TP3 – Branch-and-Cut

Cyrus Kouassi & Mahamat Abdoulaye

March 29, 2025

Question 1 – Implémentation du modèle (sans contrainte (4))

Le modèle mathématique du Problème du Voyageur de Commerce (TSP) a été codé en Python avec CPLEX. Les équations suivantes ont été respectées :

- (1) Minimise le coût total de déplacement : $\sum_{(i,j) \in A} c_{ij}x_{ij}$
- (2) Chaque sommet reçoit un unique arc entrant : $\sum_{i \neq j} x_{ij} = 1$
- (3) Chaque sommet possède un unique arc sortant : $\sum_{j \neq i} x_{ij} = 1$
- (5) Variables binaires : $x_{ij} \in \{0, 1\}$

Une fonction générique `lire_instance()` permet de charger automatiquement la matrice de coûts depuis un fichier texte, ce qui permet de tester plusieurs instances (ex. `tsp_small.txt`, `tsp_large.txt`) sans modifier le code.

Question 2 – Résolution sur `tsp_small.txt`

Le modèle est exporté avec `model.write('tsp_model.lp')` pour vérification :

- La fonction objectif est correctement définie.
- Les contraintes (2) et (3) sont correctement générées.

Résultat obtenu : La résolution du modèle sans la contrainte (4), appliqué à l’instance `tsp_small.txt`, fournit une solution optimale avec un coût total de 881.0, cette solution minimise bien la fonction objectif.

Les arcs déterminés sont : [(0, 3), (1, 4), (2, 1), (3, 0), (4, 2)]

Ces arcs forment deux cycles distincts :

- (0 → 3 → 0)
- (1 → 4 → 2 → 1)

Interprétation : bien que cette solution soit optimale d'un point de vue mathématique, elle n'est pas réalisable dans le cadre du problème du voyageur de commerce car elle ne constitue pas une tournée unique visitant tous les sommets. Il s'agit d'une solution contenant des sous-tours, rendue possible par l'absence de la contrainte (4).

Question 3 – Ajout de la contrainte (4) pour éliminer les sous-tours

La contrainte (4) a été ajoutée dynamiquement à chaque détection de sous-tours :

$$\forall S \subset V, 2 \leq |S| \leq n - 1 : \sum_{i \in S} \sum_{\substack{j \in S \\ i \neq j}} x_{ij} \leq |S| - 1$$

Ce processus se répète jusqu'à l'obtention d'une tournée unique, garantissant que la solution soit à la fois optimale et réalisable.

Tournée optimale obtenue : [0, 3, 4, 2, 1, 0]

Tous les sommets sont visités une seule fois, et le chemin forme une boucle complète revenant au point de départ. La solution satisfait donc toutes les contraintes du problème du voyageur de commerce, y compris la contrainte d'élimination des sous-tours.

Question 4 – Application à tsp_large.txt

Le même modèle a été utilisé sans modification. Seule l'instance a changé pour `tsp_large.txt`, contenant 20 sommets.

La **Tournée optimale obtenue** est :

[0, 19, 18, 16, 4, 2, 1, 8, 12, 3, 6, 9, 15, 14, 5, 13, 11, 17, 10, 7, 0]

Cette solution constitue une seule tournée fermée visitant tous les sommets une fois, et respecte donc l'ensemble des contraintes du TSP, y compris la contrainte d'élimination des sous-tours.