Livrable 1 (Check)

vendredi 11 avril 2025 11:19

Données du problème à notre livrable :

- Soit un nombre de m villes : $V = \{1, 2, \ldots, n\}$
- Coût entre deux villes i et j : $d_{ij} \in \mathbb{R}^+$

Distance

- On suppose que $d_{ii} = 0$ (pas de boucle sur elle même).

- Variables binaires de décision : $x_{ij} \in \{0; 1\}$ valant 1 si on va de i à j, sinon 0.

Objectif : trouver une combinaison x_{ij} qui fait une tournée complète. càd un cycle visitant chaque sommet.

x_q = nombre de passage dans la ville

Fonction objectif :

Minimiser la distance totale de la tournée :

$$\min \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} d_{ij} \cdot x_{ij}$$

- On compose une double somme pour énumérer tous les couples de villes (i, j) où $i \neq j$
- Ensuite pour chaque couple (i, j), on multiplie la distance d_{ij} par la variable x_{ij}.
- Si $x_{ij} = 1$, cela signifie que la tournée passe de la ville i à la ville j, donc le coût d_{ij} est compté dans la somme totale. Sinon, si $x_{ij} = 0$ alors d_{ij} est multiplié par 0 et ignoré.

Enfin, notre objectif est de minimiser la distance parcourue en assurant que chaque ville est visitée une seule fois et qu'on revient au point départ.

Contraintes :

- Chaque ville est quittée au moins 1 fois : $\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} \geq 1 \quad \forall i \in V$
- Chaque ville est atteinte au moins 1 fois : $\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ji} \geq 1 \quad \forall i \in V$
- Chaque ville quittée = atteinte : $V_i \sum_{j=1}^{} x_{ij} = \sum_{j=1}^{m} x_{ji}$

- Élimination des sous-tours (MTZ, pour $i \neq j \in \{2, \ldots, n\}$) : $u_i - u_j + n \cdot x_{ij} \leq n - 1$ Avec $u_i \in [1, n]$, des variables qui permettent de conserver l'ordre de visite. (permet d'empêcher les cycles partiels, non connectés à la tournée principale. u_i : représentant un ordre de passage dans la tournée.)

- Conservation de flux (afin d'éviter de se TP)

Contraintes ... binaires :

conservation de flux (afin d'éviter de se | P |

Contraintes supplémentaires :

time windows

Voyageur + TW → Voyageur de commerce

(intervalle infini)