Bonus : Heuristique pour le CARP

Cette heuristique est à implémenter impérativement si vous êtes en trinôme, ou optionnellement si vous êtes en binôme.

1 Bonus : problème de CARP

1.1 Rappel du problème

Données :
— un ensemble d'arcs avec une demande
— un dépôt
— une capacité unique pour les véhicules

Objectif : trouver des tournées pour les véhicules qui minimise la distance totale parcourue tout en visitant tous les arcs avec une demande.

1.2 Transformation d'un problème de CARP en problème de CVRP

Pour transformer un problème CARP en problème CVRP, on procède de la manière suivante.
On crée $3m + 1$ sommets :
— le sommet 0 (dépôt)
— trois sommets par arc (i, j) à visiter
  — s_{ij}
  — m_{ij}
  — s_{ji}

On explique maintenant comment on construit la matrice de distances $W = (w_{ij})$.
La demande de l'arc (i, j) est répartie sur les trois sommets qui le représentent de manière arbitraire.
Les distances entre les sommets sont calculées de la manière suivante. On note $\text{dist}(i, j)$ la valeur d'un chemin de plus petite valeur entre le sommet i et le sommet j dans le graphe initial, et $c(i, j)$ la distance correspondant à l'arc (i, j).

$$w(s_{ij}, s_{kl}) = \begin{cases} \frac{1}{4}\big(c(i,j) + c(k,l)\big) + \text{dist}(i,k) & \text{si } (i,j) \neq (k,l) \\ 0 & \text{sinon} \end{cases} \tag{1}$$

$$w(0, s_{ij}) = \frac{1}{4}c(i,j) + \text{dist}(i,0) \tag{2}$$

$$w(m_{ij}, v) = \begin{cases} \frac{1}{4}c(i,j) & \text{si } v = s_{ij} \text{ ou } v = s_{ji} \\ +\infty & \text{sinon} \end{cases} \tag{3}$$

1.3 Code fourni

On vous fournit un parser pour lire les instances de CARP.
Pour calculer les plus courts chemins, vous pourrez utiliser la bibliothèque de graphes de votre choix.

1.4 Travail à réaliser

Question 1. Implémentez la méthode permettant de transformer une instance de CARP en instance de CVRP.
Question 2. Testez votre méthode sur les instances fournies.