Projet RO

Badr Sajid , Mehdi Wissad

Département Sciences du Numérique - Deuxième année
2020-2021

1 Détails sur les points clés de la modélisation

1.1 Cas particulier 1 :

On veut minimiser les coûts , on modélise alors le fluide de la manière suivante . f_{ijk} comme la quantite du fluide j prise du magasin i pour la demande k.le coût est alors calculé en sommant sur i,j,k la multiplication f_{ijk} par c_{ij} le coût de chaque magasin pour chacun des fluides. on aura ici deux contraintes :

- Les quantités prises des fluides doivent être à ceux présent dans la demandes .
- Les quantités prises des fluides dans un magasin ne doit pas dépasser la quantité qui se trouve en stock.

1.2 Cas particulier 2 :

Rien de différent du premier cas , il suffit dans ce cas vu qu'on travaille avec des produits préconditionnés de rendre notre matrice de notre variable Fluides entier . En ajoutant ",Int" dans:

```
@variable(model, Fluides[1:NbF, 1:NbM , 1:NbD] >= 0, Int)
```

Avec toujours les mêmes contraintes que précédemment.

1.3 Cas particulier 3 :

On est parti sur le même code que précédemment , mais dans ce cas nous avons ajouté une nouvelle variable demande qui est un binaire afin d'ajouter les coût où pas selon si nous avons pris un colis ou pas dans un magasin . On change aussi dans ce cas notre fonction objectif. Et s'ajoute à cela deux nouvelles contraintes :

- on veut que notre demande prenne 0 où 1 selon si on a pris ou pas du fluide du magasin, demande doit alors être comprise entre sum(Fluides[:,i,j])/(sum(d[j,:])) et sum(Fluides[:,i,j])) ainsi elle ne peut prendre que 0 où 1 .

1.4 Cas particulier 4 :

Dans ce cas on veut minimiser la distance , pour cela nous avons pris une matrice de trois dimensions (Nbdem+1,Nbdem+1,Nbmag) binaire . qui pour chaque cas qui appartient à Nbmag , Matrice$Bijk = 1$ si le livreur k part du client i vers j si i,j > 1. Si i=1 le livreur par du magasin k vers le client j . si j = 1 il part du client i vers le magasin k . la fonction calcul la distance. les contraintes

2 Justification de l'adéquation du résultat

2.1 Cas particulier 1 :

la solution pour le cas particulier 1 est : coût = 9.5
En prenant les quantités suivantes , on remarque bien que de le cas 1 les valeurs ne sont pas entières. Comme dans le cas : quantite de fluide 1 1 2 = 0.5

2.2 Cas particulier 2 :

la solution pour le cas particulier 2 est : coût = 10
En prenant les quantités suivantes , on remarque bien que de le cas 2 les valeurs sont entières. le résultat est adéquat car :de fluide 1 1 2 = 1.0
Ainsi on a pas eu besoin du fluide 1 2 2 contrairement au cas 1 même si ça revenait moins cher. Ainsi le coût est logiquement plus élevé.

2.3 Cas particulier 3 :

la solution pour le cas particulier 3 est : coût = 14
le résultat est logique est notre nouvelle variable demande qui n'existait pas dans les deux cas précèdent rajoute un coût supplémentaire .

2.4 Cas particulier 4 :

la solution pour le cas particulier 4 est pour les données du fichier: "Data_test_4_2_3.txt" est :

la distance est = 2971.0

Le trajet à suivre pour le livreur du magasin 1 est :

M1 -≻ C1 -≻ C2 -≻ C4 -≻ C3 -≻ M1

Le trajet à suivre pour le livreur du magasin 2 est :

M2 -≻ C2 -≻ C4 -≻ C3 -≻ C1 -≻ M2

Le résultat est logique puisque les distance sont courtes. Par contre pour les données du fichier : "Data_test_5_2_3.txt" on retouve :

la distance est = 4133.0

Le trajet à suivre pour le livreur du magasin 1 est :

M1 -≻ C1 -≻ C2 -≻ C4 -≻ C5 -≻ C3 -≻ M1

Le trajet à suivre pour le livreur du magasin 2 est :

M2 -≻ C2 -≻ C3 -≻ C4 -≻ C5 -≻ C1 -≻ M2

3 Mise en relation qualitative entre les temps de résolution des problèmes selon le type (PL ou PLNE) et la taille des données

3.1 Cas particulier 1 :

Temps: (CPU seconds): 0.00 (Wallclock seconds): 0.04

Données:

NbF = 2 nombre de fluides disponibles

NbM = 3 nombre de magasins

NbD = 2 nombre de demandes

c = [1 2 3; 1 3 2] cout de chaque magasin

b = [2.5 1 2; 1 2 1] Stocks de fluides par magasin

d = [2 0; 1 3]

3.2 Cas particulier 2 :

Temps: Total time (CPU seconds): 0.01 (Wallclock seconds): 0.05

Données: Même données que cas 1

3.3 Cas particulier 3 :

Temps: Total time (CPU seconds): 0.01 (Wallclock seconds): 0.01

Données: Même données que cas 1 + cMag = [1 0 0 ; 0 2 1] Couts d'expédition d'un colis entre chaque paire

3.4 Cas particulier 4 :

Temps: Total time (CPU seconds): 0.01 (Wallclock seconds): 0.00

Données: Pour les données du fichier "Data_test_5_1_2.txt"

Temps: Total time (CPU seconds): 0.26 (Wallclock seconds): 0.28

Données: Pour les données du fichier "Data_test_4_2_3.txt"

Temps: Le temps est très grand

Données: Pour les données du fichier "Data_ppb_tournee_d50m10p10.1.txt"

3.5 Conclusion générale

On remarque une petite différence entre le temps de calcul entre le cas particulier 1 et 2 même si on travaille avec les même de jeux de données mais on remarque que PL (cf cas1) est plus rapide que PLNE (cf cas2) .

Pour le cas4 on ramarque que cette méthode n'est pas trop pratique pour les données volumineuses et donc il faut trouver une nouvelle version, comme utiliser les graphes pour faciliter l'étude.