Recherche Opérationnelle - TP 1

MANGENOT Guilhem, LOPEZ Léopold

Département Sciences du Numérique - Deuxième année
2023-2024

Table des matières

1 Introduction

L'objectif de ce premier TP est d'implémenter avec GLPK les problèmes linéaires d'optimisation modélisés en TD pour pouvoir ensuite les résoudre avec le solveur GLPSOL.
Nous prendrons soin de justifier nos différents choix de conception et leurs pertinences ainsi que de juger la cohérence de nos résultats vis à vis du problème énoncé.

2 Modélisation et résolution de PL/PLNE avec le solveur GLPK

2.1 Assemblage

2.1.1 Modélisation

Le sujet nous fournissant toutes les données nécessaires, nous avons modélisé ce problème au format `.lp`. Il s'agit des fichiers `Assemblage.lp.txt` et `Assemblagelp.sol.txt`.

Nous avons choisi comme variable de décision le vecteur

$$v = \begin{bmatrix} \text{quantité de vélos standards } n_s \\ \text{quantité de vélos cargos } n_c \end{bmatrix}$$

Nous choisissons trois contraintes :

- temporelle ; les ouvriers travaillent jusqu'à 60h : $0.06n_c + 0.05n_s \leq 60$
- spatiale ; le parking est limité à 1500m² : $2.5n_c + 1n_s \leq 1500$
- matérielle ; il ne faut pas assembler plus de 700 vélos cargos : $n_c \leq 700$

Et nous cherchons à maximiser la marge $700n_c + 300n_s$.

Nous avons modéliser le problème par PLNE avec $v \in \mathbb{N}^2$.

Cependant nous aurions aussi pu considérer que la production pouvait être conservée d'une semaine à l'autre et autoriser la production de vélos partiels, et la modéliser par PL. Nous aurions alors pris $v \in \mathbb{R}^2$.

Nous avons également modélisé ce problème au format gmpl pour permettre sa généralisation et nous familiariser avec le format. Ce n'était pas nécessaire.
Il s'agit des fichiers `Assemblage.mod.txt`, `Assemblagemod.lp.txt`, `Assemblagemod.sol.txt` et du fichier de données `AssemblageData1.dat.txt`.

La seule difficulté notable fut rencontrée pour généraliser la contrainte matérielle puisqu'elle ne s'applique pas à tout les types de vélos (dans l'exemple, aux vélos standards). Pour ceux-ci on pose donc que la production est inférieur à l'infini, représenté dans le code par une valeur numérique qui ne peut pas vraisemblablement être atteinte.

2.1.2 Résultats

En résolvant le problème avec les données de l'énoncé avec le solveur glpsol, et pour les deux formats, nous obtenons une marge maximum de 438.400€ en construisant 232 vélos cargos et 920 vélo standards.

Les contraintes sont respectées et le parking est pleinement exploité. Au vu des ordres de grandeur il semblerait également que ce résultat soit cohérent.

2.2 Affectaction avec prise en compte des préférences

2.2.1 Modélisation

Nous souhaitons pouvoir généraliser ce problème pour un nombre N quelconque de tâches et d'employés. Nous avons donc modélisé le problème en format gmpl.

Il s'agit des fichiers `Affectation.mod.txt`, `AffectationwithData1.lp.txt`, `AffectationwithData1.sol.txt` et des données `AffectationData1.dat.txt`.

Nous avons choisi pour variable de décision une matrice binaire d'affectation :

$$\forall (i,j) \in [|1;N|]^2 \quad \text{affectation}(i,j) = \begin{cases} 1 \text{ si l'employé } i \text{ se voit attribuer la tâche } j \\ 0 \text{ sinon} \end{cases} \quad \in [|0;1|]^{N^2}$$

Et comme donnée une matrice preferences $\in [|0,10|]^{N^2}$.

Nous avons deux contraintes garantissant que chaque ligne et colonne de la matrice affectation ne contient qu'une seule valeur 1.

Enfin, nous cherchons à maximiser la satisfaction $\sum_{i \in \text{Personnes}} \sum_{j \in \text{Tâches}} \text{affectation}(i,j) \times \text{preferences}(i,j)$.

2.2.2 Résultats

Nous avons posé arbitrairement la matrice de préférences suivante en choisissant N=5 :

$$\text{preference} = \begin{array}{c|ccccc} & T1 & T2 & T3 & T4 & T5 \\ \hline P1 & 2 & 1 & 5 & 4 & 7 \\ P2 & 1 & 1 & 1 & 1 & 1 \\ P3 & 9 & 9 & 9 & 9 & 9 \\ P4 & 5 & 8 & 2 & 3 & 5 \\ P5 & 5 & 6 & 8 & 9 & 1 \end{array}$$

Et nous obtenons les affectations suivantes :

Employé	Tâche
$P1$	$T5$
$P2$	$T1$
$P3$	$T3$
$P4$	$T2$
$P5$	$T4$

Toutes les contraintes sont respectées : nous observons que chaque employé se voit attribué une unique tâche, donc chaque tâche sera complétée une seule fois.

De plus, le bonheur de l'ensemble des employés semble bien maximisé puisque P1, P4 et P5 se voit attribué les tâches qu'ils préfèrent, tandis que P2 et P3 comblent pour satisfaire les contraintes (étant donné qu'ils n'ont aucune préférence entre les différentes tâches). Le résultat semble donc cohérent.

3 Applications en optimisation pour l'e-commerce

3.1 Premier Cas Particulier

3.1.1 Modélisation

Nous souhaitions résoudre le problème pour des nombres N_d de demandeurs, N_m de magasins et N_f de fluides, nous avons donc modélisé le problème en format gmpl.

Il s'agit des fichiers `Fluides.mod.txt`, `FluidesWithData1.lp.txt`, `FluidesWithData1.sol.txt` et des données `FluidesData1.dat.txt`.

Nous avons pris pour variable de décisions une matrice de livraison telle que

$$\text{livraison}(i,j) = \text{la quantité de fluides j livré par le magasin i}$$

Ainsi, livraison $\in \mathbb{R}+^{N_m \times N_f}$ en considérant que les quantités de fluides livrées peuvent prendre des valeurs décimales.

Nos données sont les matrices demandes $\in \mathbb{R}+^{N_d \times N_f}$, stocks $\in \mathbb{R}+^{N_f \times N_m}$ et couts $\in \mathbb{R}+^{N_f \times N_m}$.

3.1.2 Résultats

Nous avons écrit les données en reprenant les tableaux (a), (b), (c) du sujet et nous obtenons un coût minimum de 9,5€ avec la matrice de livraison suivante :

$$livraison = \begin{array}{c|cc} & F1 & F2 \\ \hline M1 & 2,5 & 1 \\ M2 & 0,5 & 1 \\ M3 & 0 & 1 \end{array}$$

Dans notre modélisation les demandes doivent être satisfaites, ainsi si nos données comprennent des demandes supérieurs à la quantité de fluides en stock alors le problème d'optimisation ne possèdent pas de solutions.

C'est le cas mis en évidence par le deuxième jeu de données posé et fourni `FluidesData2.dat.txt` et les fichiers `FluidesWithData2.lp.txt` et `FluidesWithData2.sol.txt`

3.2 Deuxième Cas Particulier

3.2.1 Modélisation

Nous modélisons toujours le problème en format gmpl. Il correspond aux fichiers `Colis.mod.txt`, `ColisWithData1.lp.txt`, `ColisWithData1.sol.txt` et aux données `ColisData1.dat.txt`.

Nous avons pris pour variables de décisions :

- La variable à 3 dimensions livraison telle que

  $$\text{livraison}(d,f,m) = \text{la quantité de fluides } f \text{ livré par le magasin } m \text{ au demandeur } d$$

  Ainsi, livraison $\in \mathbb{R}+^{Nm \times Nf \times Nd}$ (on considère encore que les quantités de fluides livrées sont réelles).
- La matrice binaire définie telle que

  $$\text{livraisonBinaire}(d,m) = \begin{cases} 1 \text{ si le magasin } m \text{ livre le demandeur } d \\ 0 \text{ sinon} \end{cases}$$

  Ainsi livraisonBinaire $\in [|0;1|]^{[|1;Nd|] \times [|1;Nm|]}$.

Nous ajoutons aux données précédentes un vecteur coutsFixes $\in \mathbb{R}+^{N_m}$ et une matrice coutsVariables $\in \mathbb{R}+^{N_d \times N_m}$.

Nous conservons les mêmes contraintes.

Nous cherchons toujours à minimiser le coût, auquel s'ajoutent les coûts fixes et variables.

3.2.2 Résultats

Pour former les données nous avons repris les tableaux (a), (b), (c) auxquels nous avons ajouté les tableaux (d) et (e) pour prendre en compte le coût du transport.

On obtient un coût minimum de 368€, et la matrice de livraison pouvant être représentée par les tableaux suivants :

$$livraison(D1) = \begin{array}{c|cc} & F1 & F2 \\ \hline M1 & 0 & 0 \\ M2 & 0 & 0 \\ M3 & 2 & 0 \end{array} \qquad livraison(D2) = \begin{array}{c|cc} & F1 & F2 \\ \hline M1 & 1 & 1 \\ M2 & 0 & 2 \\ M3 & 0 & 0 \end{array}$$

Le demandeur $D1$ se fera donc livrer par le magasin $M3$, et le demandeur $D2$ sera livré par les magasins $M1$ et $M2$.

Nos contraintes sont donc respectées puisque les demandes sont pleinement satisfaites. L'ordre de grandeur du minimum ne semblent pas abberant au vu du tableau des coups fixes.

Ainsi nos résultats semblent cohérents.

3.3 Troisième Cas Particulier

3.3.1 Modélisation

Ce cas particulier correspond au problème du voyageur de commerce vu en cours. Nous le modélisons encore au format gmpl. Il s'agit des fichiers `Livreur.mod.txt`, `LivreurWithData1.lp.txt`, `LivreurWithData1.sol.txt` et des données `LivreurData1.dat.txt` .

Pour N destinations (N-1 clients + 1 magasin) nous avons pris pour variables de décisions :

- La matrice binaire définie telle que

$$\text{matriceLivraison}(i,j) = \begin{cases} \text{1 si le livreur emprunte le chemin de } i \text{ à } j \\ \text{0 sinon} \end{cases} \quad \in [|0;1|]^{N \times N}$$

- Le vecteur ordreLivraison $\in \mathbb{N}^N$ permettant d'assurer que le livreur progresse de destination en destination en un seul cycle.

Nous prenons pour données la matrice des distances entres les destinations (entières) $\in \mathbb{N}^{N^2}$.

Nous avons deux contraintes garantissant que chaque ligne et colonne de la matrice de livraison ne contiennent qu'une seule valeur 1.

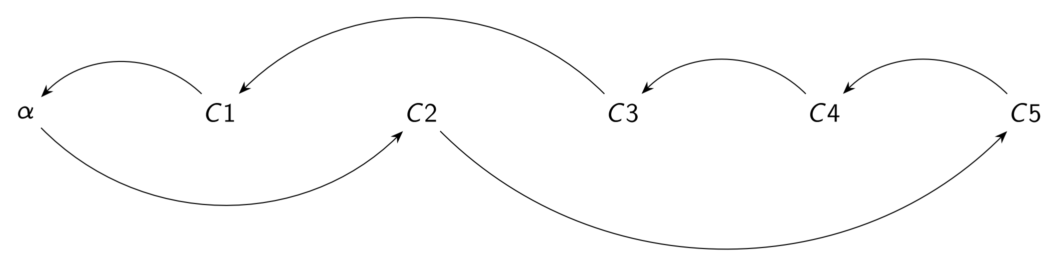
Nous cherchons à minimiser la distance totale parcourue par le livreur.

3.3.2 Résultats

Nous avons écrit les données en reprenant le tableau (f) du sujet et nous obtenons une distance minimum de 22km avec la matrice de livraison suivante :

$$\text{matriceLivraison} = \begin{array}{cc} & \begin{array}{cccccc} \alpha & C1 & C2 & C3 & C4 & C5 \end{array} \\ \begin{array}{c} \alpha \\ C1 \\ C2 \\ C3 \\ C4 \\ C5 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Cette matrice peut se traduire en cycle ainsi :



Tout d'abord, ce graphe nous permet de visualiser que nos contraintes sont respectées. De plus, les chemins empruntés sembles cohérents avec la matrice des distances puisque le livreur emprunte généralement les chemins les plus courts.

Notre résultat paraît donc vraisemblable.

4 Conclusion

Pour finir, nous avons trouvé intéressant de modéliser des problèmes réels et de les résoudre dans le cadre de ce travail pratique qui bien permis de renforcer notre compréhension de la modélisation des problèmes d'optimisation.

Cependant, nous aurions aimé avoir le temps d'essayer et de fournir un plus grand nombre de jeu de données pour pouvoir vérifier davantage les modélisations de ces problèmes, et de par exemple tester plus de cas limites.