Rapport de Recherche Opérationnelle
Modélisation et résolution de PL/PLNE avec le solveur GLPK

POLI Florian
RAGOT Cyrian

Décembre 2024

Table des matières

1 Assemblage

Pour ce problème, voir les fichiers dans le dossier PbAssemblage. Les fichiers de résolution sont PbAssemblagePL.lp.txt et PbAssemblagePLNE.lp.txt tandis que les solutions sont enregistrées dans SolAssemblagePL.sol.txt et SolAssemblagePLNE.sol.txt.

1.1 Modélisation

Nous avons choisi le même modèle (au domaine près) pour le problème linéaire et le problème linéaire en nombre entiers. La fonction objectif correspond au bénéfice produit sur une semaine.

Variables	**Fonction objectif**
$nbVC$ $nbVS$	$max(nbVC * 700 + nbVS * 300)$
Domaine	**Contraintes**
PLNE : $\mathbb{N}$ PL : $\mathbb{R}$	(a) $6 * nbVC + 5 * nbVS \leq 6000$ (b) $2.5 * nbVC + 1 * nbVS \leq 1500$ (c) $nbVC \leq 700$ (d) $nbVC, nbVS \geq 0$

$nbVC$: nombre de vélos cargos produits sur une semaine
$nbVS$: nombre de vélos standards produits sur une semaine

Justification des contraintes :

(a) Chaque semaine l'équipe fournit 60 heures de travail maximum et pour produire 100 modèles de vélos cargos il faut 6h de travail et pour 100 vélos standards il faut 5h

(b) Les vélos sont rangés chaque semaine dans un parking qui fait 1500 m^2, un vélo cargo occupe 2.5 m^2 et un vélo standard occupe 1 m^2

(c) On ne peut pas assembler plus de 700 vélos cargos par semaines

(d) Les nombres de vélos sont positifs

Nous avons choisi le format de solveur en .lp qui contient les données et le modèle car il suffit de résoudre le problème une seule fois avec les données fournies. Dans ce cas de figure, il est peu probable que l'on ai besoin de modifier les données et résoudre une nouvelle fois le problème.

1.2 Résultats

— **PL** —
Bénéfice : 438461.5385
nbVC : 230.769
nbVS : 923.077

— **PLNE** —
Bénéfice : 438400
nbVC : 232
nbVS : 920

On peut aussi remarquer que les contraintes (c) et (d) n'interviennent pas dans les résultats. On pouvait s'y attendre. En effet, pour la contrainte (d), on cherche à produire plus de vélos et un nombre de vélo n'est pas négatif sinon le problème serait mal posé. Et pour la contrainte (c), on remarque que si cette contrainte était atteinte, on aurait produit 700 vélos cargos qui auraient occupés un espace de $700 * 2.5 = 1750m^2$ et cela n'est pas possible à cause de la contrainte (b). Finalement $(b) => (c)$.

De plus, le modèle PL n'est peut être pas convaincant car il ferait faire fractions de vélos mais si on considère que l'on peut reprendre la construction d'un vélo non terminé la semaine d'après, ce modèle permet un meilleur bénéfice.

Aussi, un meilleur bénéfice avec le modèle PL est normal car ce modèle est plus permissif que le modèle PLNE.

2 Affectation avec prise en compte des préférences

Pour ce problème, voir les fichiers dans le dossier PbPreference. Les fichiers de résolution sont PbPreference.mod.txt pour le modèle et PbPreferenceData_YY.dat.txt pour les données. L'équivalent au format .lp a été généré dans le fichier PbPreference_YY.lp.txt et la solution est enregistrée dans SolPreference_YY.sol.txt.

2.1 Modélisation

Notre donnée d'entrée est une matrice carrée notée $PREF \in \mathbb{M}_N([0,10])$ d'entiers compris entre 0 et 10.
La variable est une matrice carrée de même taille notée $X \in \mathbb{M}_N([0,1])$ et qui contient des binaires. Une ligne représente une personne et une colonne représente une tâche. Ainsi, $X(i,j)$ vaut 1 si la personne i a été associée à la tâche j et vaut 0 sinon.
Une fonction objectif possible pour maximiser le bonheur des personnes est de simplement considérer la somme des notes données par chaque personnes pour la tâche qui lui est assignée. Nous l'appellerons $PreferencesFinalesTotal$.

Variables	**Fonction objectif**
$X \in \mathbb{M}_N([0,1])$	$max[\sum_{i=1}^{N}\sum_{j=1}^{N} X(i,j) * PREF(i,j)]$
Domaine	**Contraintes**
$\mathbb{M}_N([0,1])$	(a) $\forall i \quad \sum_{j=1}^{N} X(i,j) = 1$ (b) $\forall j \quad \sum_{i=1}^{N} X(i,j) = 1$

Justification des contraintes :

(a) Chaque personne doit être associée à exactement une tâche

(b) Chaque tâche doit être associée à exactement une personne

Pour ce problème, nous avons choisi le format de solveur en .mod et .dat qui permet de séparer le modèle des données. Ceci est pertinent car les données changent d'une journée à l'autre (pas le même nombre de tâches, potentiellement pas les mêmes employés et des préférences différentes chaque jour). Il suffit alors de fournir un fichier de données différent par jour et le modèle reste le même.

2.2 Résultats

Nous avons testé notre modèle avec deux tableaux de préférences en entrée :

- *PbPreferenceData_01.dat.txt*

$$PREF = \begin{pmatrix} 4 & 8 & 6 \\ 7 & 1 & 3 \\ 5 & 5 & 7 \end{pmatrix}$$

- *PbPreferenceData_02.dat.txt*

$$PREF = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Les résultats obtenus sont alors :

- *SolPreference_01.sol.txt*

$$X = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$PreferencesFinalesTotal = 22$

- *SolPreference_02.sol.txt*

$$X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$PreferencesFinalesTotal = 0$

Le premier résultat est bien cohérent avec la matrice d'entrée, le calcul de tête est facile pour une matrice 3x3 et donne bien 22, il suffit de prendre toutes les valeurs les plus grandes sur chaque ligne. Le second résultat est un test limite et donne bien 0.

Ce second résultat permet de vérifier que l'on associe bien une tâche exactement à une personne exactement, même si les préférences sont les mêmes et donnent toutes un même résultat pour la fonction objectif. Le choix des associations tâche/personne est alors aléatoire (dans le sens où on ne peut pas prédir comment a été implémenté le solveur).

Ce résultat nous a permis de nous rendre compte d'une erreur de modélisation. Nous avions mis des inégalités pour les contraintes (a) et (b) alors que des égalités étaient nécessaires.

3 Optimisation pour l'e-commerce

Pour ce problème et tous ses cas particuliers, nous utiliserons le format de solveur en .mod et .dat car les commandes changent tous les jours et il faut donc pouvoir fournir des nouvelles données pour relancer la résolution de manière régulière. Avoir des données dans un fichier séparé du modèle permet de faciliter ce changement.

3.1 Cas particulier 1

3.1.1 Modélisation

Notons D le nombre de demandes, M le nombre de magasins et F le nombre de fluides différents ou le nombre de colis différents. On modélise le problème sous forme PL si on considère des fluides et sous forme PLNE si on considère des colis. Nos données d'entrée sont 3 matrices $COMMANDES \in \mathbb{M}_{D,F}(\mathbb{R})$, $STOCKS \in \mathbb{M}_{M,F}(\mathbb{R})$ et $COUTS \in \mathbb{M}_{M,F}(\mathbb{R})$.
La variable choisi est K, une matrice cubique de $\mathbb{M}_{D,M,F}(\mathbb{N})$ ou de $\mathbb{M}_{D,M,F}(\mathbb{R})$ selon la modélisation pour colis ou fluides. $K(i,j,v)$ correspond à la portion de fluide v prise dans le magasin j pour la commande i. La fonction objectif choisie permet de minimiser le coût de livraison des commandes, on la note *cout*.

Variables	**Fonction objectif**
K	$min[\sum_{i=1}^{D}\sum_{j=1}^{M}\sum_{v=1}^{F} K(i,j,v) * COUTS(j,v)]$
Domaine	**Contraintes**
PLNE : $K \in \mathbb{M}_{D,M,F}(\mathbb{N})$ PL : $K \in \mathbb{M}_{D,M,F}(\mathbb{R})$	(a) $\sum_{j=1}^{M} K(i,j,v) = COMMANDES(i,v)$ (b) $\sum_{i=1}^{D} K(i,j,v) \leq STOCKS(j,v)$

Justification des contraintes :

(a) La quantité totale d'un fluide pris dans les différents magasins doit être égale à la quantité demandée dans la commande.

(b) La somme des demandes pour un fluide et un magasin ne doit pas dépasser le stock de ce magasin.

3.1.2 Résultats

Le modèle a été testé avec les données suivantes du fichier *PbEcomCas1Data.dat.txt*.

$$COMMANDES = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}$$

$$STOCKS = \begin{pmatrix} 2.5 & 1 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}$$

$$COUTS = \begin{pmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 2 \end{pmatrix}$$

Lorsque l'on modélise le problème avec une livraison de colis donc en problème entier on obtient 10 en coût totale ce qui cohérent puisque si on le fait à la main pour le parfum 1 on irait chercher 2 dans le magasin 1 et le dernier dans le magasin 2 (pour un coût de 4) et pour le parfum 2 : 1 dans le magasin 1, 1 dans le magasin 3 et 1 dans le magasin 2 (pour un coût de 6) et donc un coût total de 10.
Lorsque l'on modélise le problème comme un livraison de fluide, on supprime la contrainte qui veut que les valeurs soient entière et on obtient un meilleur optimum de la fonction objectif de 9.5, ce qui est cohérent car en enlevant une contrainte on a une solution plus optimiste. Dans ce cas la commande 2 va piocher dans le magasin 1 bien qu'il n'en reste que 0.5.

3.2 Cas particulier 2

3.2.1 Modélisation

Ce modèle est similaire au précédent mais change sur la fonction objectif, c'est-à-dire le coût des commandes de parfum. Dans ce problème on veut un coût fixe et un autre variable pour éviter d'aller chercher du parfum dans tout les magasins rendant le problème plus réaliste puisque cela permet de modéliser le fait que faire un trajet à un coût indépendant de ce que l'on importe.
Cependant le coût fixe dépend de si on effectue un trajet vers ce magasin, qui n'est modélisé par aucune variable précédement. Nous avons donc ajouté une variable matricielle $TRAJET$. Ainsi, $TRAJET(i,j)$ vaut 1 si on effectue un trajet pour la commande i dans le magasin j et 0 sinon. Pour modéliser linéairement cette contrainte, nous avons posé un M assez grand permettant la liberté de K si $TRAJET$ est égale à 1 et mettre à 0 K si $TRAJET$ est nulle.

Variables	**Fonction objectif**
$K, Trajet$	$min[\sum_{i=1}^{D}\sum_{j=1}^{M}\sum_{v=1}^{F} K(i,j,v) * COUTSVARIABLE(i,j) + \sum_{i=1}^{D}\sum_{j=1}^{M} TRAJET(i,j) * COUTSFIXE(i,j)]$
Domaine	**Contraintes**
PLNE : $K \in \mathbb{M}_{D,M,F}(\mathbb{N})$ $Trajet \in \mathbb{M}_{D,M}([0,1])$	(a) $\sum_{j=1}^{M} K(i,j,v) = COMMANDES(i,v)$ (b) $\sum_{i=1}^{D} K(i,j,v) \leq STOCKS(j,v)$ (c) $K(i,j,v) <= TRAJET(i,j) * 1000$

Justification des contraintes :

(a) La quantité totale d'un fluide pris dans les différents magasins doit être égale à la quantité demandée dans la commande.

(b) La somme des demandes pour un fluide et un magasin ne doit pas dépasser le stock de ce magasin.

(c) Si on prend du parfum dans le magasins j alors le trajet est diférent de 0

3.2.2 Résultats

$$COUTSFIXES = \begin{pmatrix} 110 & 90 & 100 \\ 110 & 90 & 100 \end{pmatrix}$$

$$COUTSVARIABLES = \begin{pmatrix} 10 & 1 & 5 \\ 2 & 20 & 10 \end{pmatrix}$$

Avec les mêmes matrices de paramètre que le problème précédent et les matrices de coût ci-dessus, nous trouvons un coût minimal de 354 dont l'ordre de grandeur est cohérent puisque on remarquera que pour la commande 2 il est préférable de ne pas utiliser tous les magasins mais de se restreindre à deux seulement. Comme le montre les résultats : Trajet(D2,M3) = 0

On remarquera que le problème demande d'utiliser la modélisation des colis ce qui implique des variables entières.

3.3 Cas particulier 3

3.3.1 Modélisation

Au début, nous avons pensé que ce problème était similaire au problème des tâches et préférences, ainsi il suffisait de trouver les plus courts chemins. Cependant, lorsque nous avons lancé le test, les résultats nous ont permis de nous rendre compte qu'une boucle se formait entre C1-C2-C3 et C3-C4-C5. Pour régler ce problème nous avons pensé à mettre en place une variable intermédiaire permettant de controler les cycle.