Rapport de recherche opérationnelle Modélisation+Résolution de PL/PLNE avec le solveur GLPK

Mohammed ZOUICHA
Reda ZHANI - B2

Département Sciences du Numérique - Deuxième année
2023-2024

Table des matières

Table des figures

1 Introduction

Ce projet vise la résolution de divers problèmes d'optimisation sociale. Pour ce faire, nous avons formalisé ces problèmes en utilisant le format lp et/ou gmpl pour une approche plus générique. Les résultats ont été obtenus grâce à l'utilisation de GLPK (GNU Linear Programming Kit). Ce rapport détaille les différentes approches de modélisation pour chaque problème ainsi que les solutions obtenues grâce à l'outil GLPK.

2 Assemblage

Dans le cadre de l'assemblage de vélos dans une usine, nous avons formulé un problème d'optimisation linéaire (PL) pour maximiser la marge totale, en tenant compte de diverses contraintes.

2.1 Modèle PL :

- **Variables** : On prend Q1 et Q2 les variables de notre problème qui correspondent à la production de chaque modèle durant une semaine

- **Objectif** : Maximiser la marge totale représentée par la formule : 700Q1 + 300Q2.
  Sous réserve de :

  - **Capacité de Temps** : L'équipe ne peut pas dépasser 60 heures de travail par semaine, exprimé par l'équation 6Q1 + 5Q2 <= 6000.

  - **Capacité du Parking** : Les vélos assemblés ne doivent pas occuper plus de 1500m2 de parking, avec 2.5m2 par vélo cargo (Q1) et 1m2 par vélo standard (Q2), représenté par l'équation 2.5Q1 + Q2 <= 1500.

  - **Capacité de Batteries** : Il ne peut être assemblé plus de 700 vélos cargos (Q1) en raison de la limitation des ressources en batteries, exprimé par l'équation Q1 <= 700.

2.2 Résolution :

Nous avons réalisé le modèle de résolution sous le fichier PbAssemblagePL.lp.txt.Suite à la résolution du modèle PL, la solution optimale est atteinte avec un bénéfice maximum de 438461.5385 euros. Les quantités optimales dassemblage sont Q1 = 230.769 vélos cargos et Q2 = 923.077 vélos standards.
La figure 1 correspond aux solutions renvoyées grâce à la résolution.

```
Non-zeros:    7
Status:     OPTIMAL
Objective:  Benefice = 438461.5385 (MAXimum)

   No.   Row name   St   Activity     Lower bound   Upper bound
Marginal
------ ------------ -- ------------- ------------- -------------
-------------
     1 CapacitedeTemps
                    NU          6000                        6000
7.69231
     2 CapaciteduParking
                    NU          1500                        1500
261.538
     3 CapacitedeBatteries
                    B        230.769                         700
     4 MinQ1        B        230.769             0
     5 MinQ2        B        923.077             0

   No. Column name  St   Activity     Lower bound   Upper bound
Marginal
------ ------------ -- ------------- ------------- -------------
-------------
     1 Q1           B        230.769             0
     2 Q2           B        923.077             0

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
```

FIGURE 1 – Résultat en réel flottant du problème Assemblage

2.3 Modèle PLNE :

Dans le modèle PNLE, les variables Q1 et Q2 sont déclarées comme des variables entières. Cela caractérise le modèle comme un problème de Programmation Linéaire en Nombres Entiers (PLNE). Les contraintes sur les capacités de temps, de parking et de batteries restent les mêmes que dans le modèle PL.
Le résultat du fichier PbAssemblagePNLE.pl.txt est le suivant, on obtient un benifice maximal de valeur de 438400 euros. Les quantités optimales dassemblage sont Q1 = 232 vélos cargos et Q2 = 920 vélos standards.
:

```
Columns:    2 (2 integer, 0 binary)
Non-zeros:  5
Status:     INTEGER OPTIMAL
Objective:  Benefice = 438400 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 CapacitedeTemps
                                5992                        6000
     2 CapaciteduParking
                                1500                        1500
     3 CapacitedeBatteries
                                 232                         700

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 Q1           *            232             0
     2 Q2           *            920             0

Integer feasibility conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

FIGURE 2 – Résultat en entiers naturels du problème Assemblage

3 Affectation avec prise en compte des préférences

La gestion efficace des équipes nécessite une planification judicieuse des tâches en fonction des préférences individuelles. Dans ce contexte, une manageuse doit attribuer N tâches à N membres d'une équipe, en tenant compte des scores de préférence individuels notés sur 10. L'objectif est de maximiser la satisfaction globale de l'équipe en élaborant un modèle de Programmation Linéaire en Nombres Entiers (PLNE) pour trouver la meilleure affectation possible.

3.1 Modélisation :

- **Constantes** : La matrice des préférences $MatriceDePreference_{ij}$

- **Variables** : Nous avons opté pour l'utilisation d'une variable, notée Q, sous forme d'une matrice de taille $n \times n$ (n personnes pour n taches) binaire.
  Dans cette notation, $Q_{ij} = 1$ si la personne i s'occupe de la tache j , et $Q_{ij} = 0$ sinon. En appliquant ce concept à un modèle de Programmation Linéaire en Nombres Entiers (PLNE), le problème formulé est le suivant :

- **L'objectif** est de maximiser : $\sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{c}_{ij} \times Q_{ij}$

  Sous réserve de deux contraintes :

  - **Contrainte 1** : Chaque personne doit être affectée exactement à une tâche, exprimé par : $\forall i \in 1; n, \sum_{j=1}^{n} Q_{i,j} = 1$

  - **Contrainte 2** : Chaque tâche doit être effectuée exactement une fois, exprimé par : $\forall j \in 1; n, \sum_{i=1}^{n} Q_{i,j} = 1$

3.2 Resolution :

Pour cette étude, nous avons réalisé le modèle de résolution sous le fichier PbAffectationwith-Data.mod.txt.
Grâce à cette modélisation (et ce format de fichier), on peut appliquer différents cas, avec différents jeux de données. On a alors créé un cas avec trois personnes et trois postes.
La figure 2 correspond aux solutions renvoyées grâce à la résolution. Un score maximum de valeur 21

```
Objective:  Affectation = 21 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 RespectTacheAffectee[T1]
                                  1             1             =
     2 RespectTacheAffectee[T2]
                                  1             1             =
     3 RespectTacheAffectee[T3]
                                  1             1             =
     4 RespectPersonneAffectee[P1]
                                  1             1             =
     5 RespectPersonneAffectee[P2]
                                  1             1             =
     6 RespectPersonneAffectee[P3]
                                  1             1             =
     7 Affectation               21

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 Q[P1,T1]     *             0             0             1
     2 Q[P2,T1]     *             1             0             1
     3 Q[P3,T1]     *             0             0             1
     4 Q[P1,T2]     *             1             0             1
     5 Q[P2,T2]     *             0             0             1
     6 Q[P3,T2]     *             0             0             1
     7 Q[P1,T3]     *             0             0             1
     8 Q[P2,T3]     *             0             0             1
     9 Q[P3,T3]     *             1             0             1
```

FIGURE 3 – Résultat de l'affectation des taches aux personnes

4 Applications en optimisation pour le-commerce

Dans le domaine de l'e-commerce, l'optimisation des affectations de commandes aux magasins constitue un défi majeur. Cette optimisation repose sur des considérations financières et environnementales liées à la livraison, à la préparation des commandes, et à la gestion des stocks.
Notre focus se concentre spécifiquement sur le problème d'affectation de commandes et de planification de tournées de véhicules pour plusieurs magasins d'une enseigne ou d'une franchise, visant à minimiser les coûts et l'impact environnemental.

4.1 Cas particulier 1.1 :

4.1.1 Modélisation :

Les composantes clés du modèle comprennent :

- **Les ensembles** : DEMANDES, FLUIDES, MAGAZINS. Dans la suite de ce rapport, nous les noterons respectivement D, F et M.

- **Les variables** : Q_{ijk} représentant l'allocation des demandes aux magasins pour des fluides spécifiques.

- **Les constantes** : Matrices pour les demandes de fluides (MatriceFluidesDemandes), les stocks (MatriceStocks), et les coûts (MatriceDeCout).

- **La fonction objective** visant à minimiser le coût global associé à l'allocation des demandes, notée Cout, est définie comme la somme des coûts individuels de chaque allocation (Q_{ijk}) multipliée par le coût spécifique associé à cette allocation, tels que :

$$\text{cout} = \min \sum_{i \in \text{M}} \sum_{j \in \text{F}} \sum_{k \in \text{D}} Q[i,j,k] \cdot \text{MatriceDeCout}[i,j]$$

Sous la reserve des contraintes suivantes :

- **Respect Commande** : Cette contrainte assure que la somme des allocations des demandes pour chaque fluide j par chaque magasin i est égale à la demande spécifiée dans la matrice MatriceFluidesDemandes pour la demande k.

  $\forall j \in 1; N_{\text{F}}\ , \forall k \in 1; N_{\text{D}}, \sum_{i \in \text{M}} Q_{i,j,k} = MatriceFluidesDemande[k,j]$

- **Respect du stocks** : Cette contrainte s'assure que la somme des allocations des demandes pour chaque fluide j par chaque magasin i ne dépasse pas les niveaux de stock spécifiés dans la matrice MatriceStocks. Elle garantit ainsi que les niveaux de stock dans chaque magasin ne sont pas dépassées.

  $\forall i \in 1; N_{\text{M}}\ , \forall j \in 1; N_{\text{F}}\ , \sum_{i \in \text{D}} Q_{i,j,k} \leq MatriceStocks[i,j]$

4.1.2 Résolution :

Nous avons réalisé le modèle de résolution sous le fichier EcomCas11.mod.txt. Le résultat obtenu est dans la figure suivante :

```
   No. Column name  St   Activity     Lower bound   Upper bound
Marginal
------ ------------ -- ------------- ------------- -------------
-------------
     1 Q[M1,F1,D1]  B             2             0
     2 Q[M2,F1,D1]  NL            0             0
< eps
     3 Q[M3,F1,D1]  NL            0
0                      1
     4 Q[M1,F1,D2]  B           0.5             0
     5 Q[M2,F1,D2]  B           0.5             0
     6 Q[M3,F1,D2]  NL            0
0                      1
     7 Q[M1,F2,D1]  NL            0
0                      3
     8 Q[M2,F2,D1]  NL            0
0                      3
     9 Q[M3,F2,D1]  NL            0
0                      3
    10 Q[M1,F2,D2]  B             1             0
    11 Q[M2,F2,D2]  B             1             0
    12 Q[M3,F2,D2]  B             1             0

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
```

Figure 4 – Résultat du cas 1-1 de le-commerce

4.2 Cas particulier 1.2 :

Le cas 1.2 est similaire au premier cas, par contre le coût résultant comprend un coût fixe et un coût variable dépendant de la quantité transportée.

4.2.1 Modélisation :

Variables : Une nouvelle variable binaire *Livraison[i,j]* a été introduite, indiquant si le magasin j expédie un colis au client i.

Constantes : On introduit deux nouvelles matrices : MatriceCoutFixe[i,j] et MatriceCoutVariable[i,j] qui representent les coûts fixes associés à la livraison et les coûts variables liés à la livraison respectivement.

La fonction objective visant à minimiser le coût global est sous la forme suivante :

$$\text{Cout} = \left(\sum_{i\in \text{M}} \sum_{j\in \text{F}} \sum_{k\in \text{D}} Q[i,j,k] \cdot \text{MatriceDeCout}[i,j]\right)$$
$$+ \left(\sum_{i\in \text{D}} \sum_{j\in \text{M}} \text{Livraison}[i,j] \cdot (\text{MatriceCoutVariable}[i,j] + \text{MatriceCoutFixe}[i,j])\right)$$

Sous la reserve en plus des contraintes du cas 1 la contrainte suivante :

$$\forall i \in 1; N_{\text{magasins}}\ , \forall j \in 1; N_{\text{demandes}}\ , \sum_{k=1}^{N_{\text{fluides}}} Q_{i,j,k} > 0 \Rightarrow \text{L}\,ivraison[j,k] = 1$$

Pour exprimer cette condition sous forme d'une contrainte dans ce problème d'optimisation, on va utiliser la méthode "**Big M**". On obtient alors la contrainte suivante :

$\forall$i $\in$1 ; $\text{N}_{\text{magasins}}$, $\forall j \in 1; N_{\text{demandes}}$, $\sum_{k=1}^{N_{\text{fluides}}} Q_{i,j,k} < M \times \mathit{Livraison}[i,j]$

Avec M une constante suffisament grande. Pour determiner sa valeur, il faut généralement choisir une valeur suffisament grande pour qu'elle n'affecte pas la solution du problème d'optimisation, mais pas trop grande pour eviter les problèmes numériques. Alors, on va se baser sur les valeurs maximales possibles de (g(x) = $\sum_{k=1}^{N_{\text{fluides}}} Q_{i,j,k}$)

On a d'après la contrainte **Respect du stocks** :

$\forall i \in 1; N_{\text{M}}\ , \forall j \in 1; N_{\text{F}}\ , \sum_{k=1}^{N_{\text{D}}} Q_{i,j,k} \leq MatriceStocks[i,j]$ avec : $\text{Q}_{i,j,k} \geq 0$

Alors : $Q_{i,j,k}$ $\leq$MatriceStocks[i, j]

Donc : $Q_{i,j,k}$ $\leq$max(MatriceStocks[i, j])

En introduisant la somme du k in FLUIDES, on obtient :

$$\forall i \in 1; N_{\text{D}}\ , \forall j \in 1; N_{\text{M}}\ , \sum_{k=1}^{N_{\text{F}}} Q_{j,k,i} \leq max(MatriceStocks[i,j]) \times Card(F)$$

On choisit donc M = max(MatriceStocks[i, j]) $\times$Card(FLUIDES), on obtient la contrainte suivante :

Contrainte Binaire :

$\sum_{k=1}^{N_F} Q_{i,j,k} < max(MatriceStocks[i,j]) \times M$

4.2.2 Résolution :

Pour le troisième cas, nous avons réalisé le modèle de résolution sous le fichier EcomCas12.mod.txt. Nous avons appliquer ce modèle sur le même jeu de donnée que précemment en ajoutant celui des coûts fixes et variables présent dans lénoncé du projet. Le résultat obtenu est dans la figure 5.

```
   17 Cout                      341

  No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 Q[M1,F1,D1]                0             0
     2 Q[M2,F1,D1]                0             0
     3 Q[M3,F1,D1]                2             0
     4 Q[M1,F1,D2]                1             0
     5 Q[M2,F1,D2]                0             0
     6 Q[M3,F1,D2]                0             0
     7 Q[M1,F2,D1]                0             0
     8 Q[M2,F2,D1]                0             0
     9 Q[M3,F2,D1]                0             0
    10 Q[M1,F2,D2]                1             0
    11 Q[M2,F2,D2]                2             0
    12 Q[M3,F2,D2]                0             0
    13 Livraison[D1,M1]
                    *             0             0             1
    14 Livraison[D1,M2]
                    *             0             0             1
    15 Livraison[D1,M3]
                    *             1             0             1
    16 Livraison[D2,M1]
                    *             1             0             1
    17 Livraison[D2,M2]
                    *             1             0             1
    18 Livraison[D2,M3]
                    *             0             0             1
```

FIGURE 5 – Résultat du cas 1-2 de le-commerce

4.3 Cas particulier 2 :

Dans ce cas, nous nous intéressons à la livraison vers différents clients à partir dun magasin. Le livreur doit parcourir tout les clients à la suite avant de revenir au magasin. En sachant les distances séparant les différents points de remise, le probleme est de minimiser la distance parcourue par le livreur. Ce problème correspond au problème classique du Voyageur de Commerce.

4.3.1 Modélisation :

Les composantes clés du modèle comprennent :
Ensembles : ALPHAETCLIENT (représente l'ensemble des lieux, comprenant le point de départ "ALPHA" et les différents clients "C1", "C2", etc)

Variables :

- Q_{ij} une variable binaire indiquant si il y une liaison de i vers j dans la tournée du voyageur de commerce.

- U_i est une variable entière utilisée pour éviter la formation de sous-tours dans la formulation de MillerTuckerZemlin.
  En effet, En plus des Q_{ij} variables ci-dessus, il y a pour chacune i = [1,n] ,une variable fictive

U_i qui suit l'ordre dans lequel les lieux sont visitées, à compter du lieux 1 ; l'interprétation est que $U_i < U_j$ implique que le lieu i est visitée avant le lieu j.

Constantes : MatriceDesDistances[i,j] représente les distances entre les lieux i et j

La fonction objective visant à minimiser la distance totale parcourue par le voyageur de commerce, représentée sous la forme :

$\sum_{i=1}^{N_{\text{ALPHAETCLIENT}}} \sum_{j=1}^{N_{\text{ALPHAETCLIENT}}}$ Q_{ij} $\times$MatriceDesDistances[i,j]

Sous la reserve des contraintes suivantes :

RespectVisiter : Chaque lieu doit être visité une fois.

$\forall$j $\in$1 ; N$_{\text{ALPHAETCLIENT}}$,$\sum_{i\neq j,i=1}^{N_{\text{ALPHAETCLIENT}}}$ $Q_{i,j} = 1$

RespectQuitter : Chaque lieu doit être quitté une fois.

$\forall$i $\in$1 ; N$_{\text{ALPHAETCLIENT}}$,$\sum_{j\neq i,j=1}^{N_{\text{ALPHAETCLIENT}}}$ $Q_{i,j} = 1$

EliminationSousTours : Contrainte de MillerTuckerZemlin pour éliminer les sous-tours dans la solution.

$$U_i - U_j + N_{\text{ALPHAETCLIENT}}\ Q_{ij} \quad \leq n - 1 \qquad 2 \leq i \neq j \leq n;$$

.

UneSeuleTour : Assure que chaque lieu est inclus dans une et une seule tournée.
$1 \leq U_i \leq n \qquad 1 \leq i \leq n;$

4.3.2 Résolution :

Avec ce modèle, nous obtenons, en appliquant les données du sujet écrites dans le fichier EcomCas2.dat.txt, une distance total de 22.

Les résultats sont disponibles dans la figure 5.

No.	Column name		Activity	Lower bound	Upper bound
1	Q[C1,ALPHA]	*	1	0	1
2	Q[C2,ALPHA]	*	0	0	1
3	Q[C3,ALPHA]	*	0	0	1
4	Q[C4,ALPHA]	*	0	0	1
5	Q[C5,ALPHA]	*	0	0	1
6	Q[ALPHA,C1]	*	0	0	1
7	Q[C2,C1]	*	0	0	1
8	Q[C3,C1]	*	1	0	1
9	Q[C4,C1]	*	0	0	1
10	Q[C5,C1]	*	0	0	1
11	Q[ALPHA,C2]	*	1	0	1
12	Q[C1,C2]	*	0	0	1
13	Q[C3,C2]	*	0	0	1
14	Q[C4,C2]	*	0	0	1
15	Q[C5,C2]	*	0	0	1
16	Q[ALPHA,C3]	*	0	0	1
17	Q[C1,C3]	*	0	0	1
18	Q[C2,C3]	*	0	0	1
19	Q[C4,C3]	*	1	0	1
20	Q[C5,C3]	*	0	0	1
21	Q[ALPHA,C4]	*	0	0	1
22	Q[C1,C4]	*	0	0	1
23	Q[C2,C4]	*	0	0	1
24	Q[C3,C4]	*	0	0	1
25	Q[C5,C4]	*	1	0	1
26	Q[ALPHA,C5]	*	0	0	1
27	Q[C1,C5]	*	0	0	1
28	Q[C2,C5]	*	1	0	1
29	Q[C3,C5]	*	0	0	1
30	Q[C4,C5]	*	0	0	1

FIGURE 6 – Résultat du cas 2 de le-commerce