Recherche Opérationnelle

Rapport TP1

2SN L4

Élèves :

THEVENET Louis

SABLAYROLLES Guillaume

9 Décembre 2024

Table des matières

1 Modélisation et Résolution de PL/PLNE avec le solveur GLPK 3
1.1 Assemblage 3
1.2 Applications en optimisation pour l'e-commerce 5

1 Modélisation et Résolution de PL/PLNE avec le solveur GLPK

1.1 Assemblage

Ce problème peut se modéliser par PL dans le cas où la fabrication interrompue en fin de semaine d'un vélo peut être continuée en début de la semaine suivante. Au contrainte, si on est obligé de fabriquer les nouveaux vélos de zéro chaque semaine, le problème se modélise par PLNE.

Variables

Nombre de vélos cargos $C \in \mathbb{R}^+$ (ou entière dans le cas PLNE)
Nombre de vélos cargos $S \in \mathbb{R}^+$ (ou entière dans le cas PLNE)

Fonction objectif

$$f(C, S) = \max(700C + 300S)$$

Contraintes

Respect du nombre d'heures $0.06C + 0.05S \leq 60$
Respect de la surface maximale occupée $2.5C + 1S \leq 1500$
Respect du nombre max de vélos cargos produits $C \leq 700$

Solution PLNE

```
Problem:
Rows:       3
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 TravailHebdo            59.92                          60
     2 SurfaceOccupee
                                1500                        1500
     3 ProductionCargoMax
                                 232                         700

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 C            *            232             0
     2 S            *            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
```

On constate que la solution trouvée $(C, S) = (232, 920)$ maximise l'objectif avec $f(C, S) = 438400$€. Le nombre d'heures nécessaires pour ce résultat est 59.92h et la surface disponible est complètement utilisée. Si on augmente la surface disponible, on peut alors produire plus de vélos, on peut également faire varier le ratio $\frac{\text{Place occupée par un vélo cargo}}{\text{Place occupée par un vélo normal}}$, ce qui permettrait de produire plus de vélos cargo (la limite de 700 n'est pas atteinte car ce n'est pas « rentable » de faire des cargos avec ces paramètres.).

1.1.1 Affectation avec prise en compte des préférences

Données

$\boldsymbol{n} \in \mathbb{N}$ nombre de personnes
$\boldsymbol{m} \in \mathbb{N}$ nombre d'activités
$\boldsymbol{P} \in \mathcal{M}_{\boldsymbol{n},\boldsymbol{m}}(\mathbb{R})$ Matrice des préférences

Variables

On utilise une matrice $M \in \mathcal{M}_{n,m}(\mathbb{R})$ avec n le nombre de personnes, m le nombres d'activités, telle que

$$\forall 1 \leq i \leq n \forall 1 \leq j \leq m M_{i,j} = \begin{cases} 1 \text{ si la personne } i \text{ réalise l'activité } j \\ 0 \text{ sinon} \end{cases}$$

Fonction objectif

$$f : \begin{cases} \mathcal{M}_{n,m}(\mathbb{R}) \to \mathbb{R} \\ M \qquad \mapsto \sum_{i=1}^{n} \sum_{j=1}^{m} M_{i,j} \times P_{i,j}) \end{cases}$$

où P est la matrice des préférences, une donnée du problème.

Contraintes

Une personne est associée à une seule activité $\forall 1 \leq i \leq m \sum_{j=1}^{m} M_{i,j} = 0$

Une activité est associée à une seule personne $\forall 1 \leq j \leq m \sum_{i=1}^{n} M_{i,j} = 0$

Solution

Pour $P = \begin{pmatrix} 9 & 5 & 1 \\ 2 & 4 & 2 \\ 9 & 4 & 8 \end{pmatrix}$, la solution trouvée est $M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. On vérifie aisément que c'est la solution optimale.

```
Problem:    PbPreferences
Rows:       7
Columns:    9 (9 integer, 9 binary)
Non-zeros:  27
Status:     INTEGER OPTIMAL
Objective:  SatisfactionTotale = 21 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 RespectDistributionLigne[P1]
                                   1             1             =
     2 RespectDistributionLigne[P2]
                                   1             1             =
     3 RespectDistributionLigne[P3]
                                   1             1             =
     4 RespectDistributionColonne[T1]
                                   1             1             =
     5 RespectDistributionColonne[T2]
                                   1             1             =
```

```
     6 RespectDistributionColonne[T3]
                                  1             1             =
     7 SatisfactionTotale
                                 21

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

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
```

1.2 Applications en optimisation pour l'e-commerce

1.2.1 Cas particulier 1.1

Données

$\boldsymbol{f} \in \mathbb{N}$ nombre de fluides
$\boldsymbol{m} \in \mathbb{N}$ nombre de magasins
$\boldsymbol{d} \in \mathbb{N}$ nombre de demandes

Et trois matrices:

- fluides_par_demandes $\in \mathcal{M}_{d,f}(\mathbb{R})$
- stock_par_magasin $\in \mathcal{M}_{m,f}(\mathbb{R})$
- cout_par_magasin $\in \mathcal{M}_{m,f}(\mathbb{R})$

Variables

On utilise une matrice $D \in \mathcal{M}_{f,m,d}(\mathbb{R})$ avec

- f le nombre de fluides différents
- m le nombre de magasins
- d le nombre de demandes réalisées telle que

$$\forall 1 \leq i \leq f \forall 1 \leq j \leq m \forall 1 \leq k \leq d,$$
$$D_{i,j,k} \text{ est la quantité de fluide } i \text{ demandée au magain } j \text{ lors de la demande } k$$

Fonction objectif

$$f : \begin{cases} \mathcal{M}_{f,m,d}(\mathbb{R}) \to \mathbb{R} \\ D \qquad\qquad \mapsto \sum_{i=1}^{f} \sum_{i=1}^{m} \sum_{k=1}^{d} \text{cout\_par\_magasin}_{j,i} D_{i,j,k} \end{cases}$$

Contraintes

Le nombre total d'un fluide des demandes ne dépasse pas les stocks $\forall 1 \leq i \leq f, \forall 1 \leq j \leq m \sum_{k=1}^{d} D_{i,j,k} \leq \text{stock\_par\_magasin}_{j,i}$

Les fluides par demande sont respectés $\forall 1 \leq i \leq f, \forall 1 \leq k \leq d \sum_{j=1}^{m} D_{i,j,k} = \text{fluides\_par\_demandes}_{k,i}$

Solution

Pour $\text{fluides\_par\_demandes} = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}$, $\text{stock\_par\_magasin} = \begin{pmatrix} 2.5 & 1 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}$ et $\text{cout\_par\_magasin} = \begin{pmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 2 \end{pmatrix}$, la solution pour un coût minimum est de : $\text{CoutTotal} = 9.5$ pour la matrice $D = [\text{D1}, \text{D2}]$ avec $\text{D1} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ et $\text{D2} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

```
Problem:    PbMagasin
Rows:       11
Columns:    12
Non-zeros:  36
Status:     OPTIMAL
Objective:  CoutTotal = 9.5 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 RespectStock[F1,M1]
                    NU           2.5                         2.5            -1
     2 RespectStock[F1,M2]
                    B            0.5                           1
     3 RespectStock[F1,M3]
                    B              0                           2
     4 RespectStock[F2,M1]
                    NU             1                           1            -2
     5 RespectStock[F2,M2]
                    B              1                           2
     6 RespectStock[F2,M3]
                    NU             1                           1            -1
     7 RespectDemande[F1,D1]
                    NS             2             2             =             2
     8 RespectDemande[F1,D2]
                    NS             1             1             =             2
     9 RespectDemande[F2,D1]
                    B              0            -0             =
    10 RespectDemande[F2,D2]
                    NS             3             3             =             3
    11 CoutTotal    B            9.5

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 D[F1,M1,D1]  B              2             0
     2 D[F1,M1,D2]  B            0.5             0
     3 D[F1,M2,D1]  NL             0             0                       < eps
     4 D[F1,M2,D2]  B            0.5             0
     5 D[F1,M3,D1]  NL             0             0                           1
```

```
     6 D[F1,M3,D2]  NL             0             0                           1
     7 D[F2,M1,D1]  NL             0             0                           3
     8 D[F2,M1,D2]  B              1             0
     9 D[F2,M2,D1]  NL             0             0                           3
    10 D[F2,M2,D2]  B              1             0
    11 D[F2,M3,D1]  NL             0             0                           3
    12 D[F2,M3,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
        High quality

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

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

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

End of output
```

1.2.2 Cas particulier 1.2

Données

$\boldsymbol{f} \in \mathbb{N}$ nombre de fluides
$\boldsymbol{m} \in \mathbb{N}$ nombre de magasins
$\boldsymbol{d} \in \mathbb{N}$ nombre de demandes

Et cinq matrices:

- fluides_par_demandes $\in \mathcal{M}_{d,f}(\mathbb{R})$
- stock_par_magasin $\in \mathcal{M}_{m,f}(\mathbb{R})$
- cout_par_magasin $\in \mathcal{M}_{m,f}(\mathbb{R})$
- cout_fixe $\in \mathcal{M}_{d,m}(\mathbb{R})$
- cout_variable $\in \mathcal{M}_{d,m}(\mathbb{R})$

Variables

On utilise une matrice $D \in \mathcal{M}_{f,m,d}(\mathbb{R})$ avec

- f le nombre de fluides différents
- m le nombre de magasins
- d le nombre de demandes réalisées telle que

$$\forall 1 \leq i \leq f \forall 1 \leq j \leq m \forall 1 \leq k \leq d,$$
$$D_{i,j,k} \text{ est la quantité de fluide } i \text{ demandée au magain } j \text{ lors de la demande } k$$

Fonction objectif

$$f : \begin{cases} \mathcal{M}_{f,m,d}(\mathbb{R}) \to \mathbb{R} \\ D \qquad\qquad \mapsto \sum_{i=1}^{f} \sum_{i=1}^{m} \sum_{k=1}^{d} (\text{cout\_par\_magasin}_{j,i} + \text{cout\_variable}_{k,j}) D_{i,j,k} + \text{cout\_fixe}_{k,j} \end{cases}$$

Contraintes

Le nombre total d'un fluide des demandes ne dépasse pas les stocks $\forall 1 \leq i \leq f, \forall 1 \leq j \leq m \sum_{k=1}^{d} D_{i,j,k} \leq \text{stock\_par\_magasin}_{j,i}$

Les fluides par demande sont respectés $\forall 1 \leq i \leq f, \forall 1 \leq k \leq d \sum_{j=1}^{m} D_{i,j,k} = \text{fluides\_par\_demandes}_{k,i}$

Solution

Pour $\text{fluides\_par\_demandes} = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}$, $\text{stock\_par\_magasin} = \begin{pmatrix} 2.5 & 1 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}$, $\text{cout\_par\_magasin} = \begin{pmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 2 \end{pmatrix}$, $\text{cout\_fixe} = \begin{pmatrix} 110 & 90 & 100 \\ 110 & 90 & 100 \end{pmatrix}$ et $\text{cout\_variable} = \begin{pmatrix} 10 & 1 & 5 \\ 2 & 20 & 10 \end{pmatrix}$, la solution pour un coût minimum est de : $\text{CoutTotal} = 1252$ pour la matrice $D = [\text{D1}, \text{D2}]$ avec $\text{D1} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$ et $\text{D2} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

```
Problem:    Pb2
Rows:       11
Columns:    12
Non-zeros:  36
Status:     OPTIMAL
Objective:  CoutTotal = 1252 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 RespectStock[F1,M1]
                    B              1                         2.5
     2 RespectStock[F1,M2]
                    NU             1                           1            -5
     3 RespectStock[F1,M3]
                    B              1                           2
     4 RespectStock[F2,M1]
                    NU             1                           1           -20
     5 RespectStock[F2,M2]
                    B              1                           2
     6 RespectStock[F2,M3]
                    NU             1                           1           -11
     7 RespectDemande[F1,D1]
                    NS             2             2             =             8
     8 RespectDemande[F1,D2]
                    NS             1             1             =             3
     9 RespectDemande[F2,D1]
                    B              0            -0             =
    10 RespectDemande[F2,D2]
                    NS             3             3             =            23
    11 CoutTotal    B             52

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 D[F1,M1,D1]  NL             0             0                           3
     2 D[F1,M1,D2]  B              1             0
     3 D[F1,M2,D1]  B              1             0
     4 D[F1,M2,D2]  NL             0             0                          24
     5 D[F1,M3,D1]  B              1             0
     6 D[F1,M3,D2]  NL             0             0                          10
     7 D[F2,M1,D1]  NL             0             0                          31
     8 D[F2,M1,D2]  B              1             0
     9 D[F2,M2,D1]  NL             0             0                           4
    10 D[F2,M2,D2]  B              1             0
```

```
    11 D[F2,M3,D1]  NL             0             0                          18
    12 D[F2,M3,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
        High quality

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

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

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

End of output
```

1.2.3 Cas particulier 2