



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 contraire, 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

```
1 Problem:
2 Rows:      3
3 Columns:    2 (2 integer, 0 binary)
4 Non-zeros:  5
5 Status:     INTEGER OPTIMAL
6 Objective:  Benefice = 438400 (MAXimum)
7
8   No.  Row name      Activity  Lower bound  Upper bound
9   ----  -
10    1 TravailHebdo    59.92
11    2 SurfaceOccupee  1500
12    3 ProductionCargoMax  700
13
14   No.  Column name  Activity  Lower bound  Upper bound
15   ----  -
16    1 C              *    232          0
17    2 S              *    920          0
18
19 Integer feasibility conditions:
20
21 KKT.PE: max.abs.err = 0.00e+00 on row 0
22         max.rel.err = 0.00e+00 on row 0
23         High quality
24
25 KKT.PB: max.abs.err = 0.00e+00 on row 0
26         max.rel.err = 0.00e+00 on row 0
27         High quality
28
29 End of output
```

On constate que la solution trouvée $(C, S) = (232, 920)$ maximise l'objectif avec $f(C, S) = 438400\text{€}$. 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

$n \in \mathbb{N}$ nombre de personnes

$m \in \mathbb{N}$ nombre d'activités

$P \in \mathcal{M}_{n,m}(\mathbb{R})$ 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

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

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

Contraintes

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

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

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.

```

1 Problem:      PbPreferences
2 Rows:        7
3 Columns:     9 (9 integer, 9 binary)
4 Non-zeros:   27
5 Status:      INTEGER OPTIMAL
6 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	=

```

20      6 RespectDistributionColonne[T3]
21          1          1          =
22      7 SatisfactionTotale
23          21
24
25      No. Column name      Activity      Lower bound      Upper bound
26      -----
27      1 M[P1,T1]      *          1          0          1
28      2 M[P1,T2]      *          0          0          1
29      3 M[P1,T3]      *          0          0          1
30      4 M[P2,T1]      *          0          0          1
31      5 M[P2,T2]      *          1          0          1
32      6 M[P2,T3]      *          0          0          1
33      7 M[P3,T1]      *          0          0          1
34      8 M[P3,T2]      *          0          0          1
35      9 M[P3,T3]      *          1          0          1
36
37      Integer feasibility conditions:
38
39      KKT.PE: max.abs.err = 0.00e+00 on row 0
40              max.rel.err = 0.00e+00 on row 0
41              High quality
42
43      KKT.PB: max.abs.err = 0.00e+00 on row 0
44              max.rel.err = 0.00e+00 on row 0
45              High quality
46
47      End of output

```

1.2 Applications en optimisation pour l'e-commerce

1.2.1 Cas particulier 1.1

Données

$f \in \mathbb{N}$ nombre de fluides

$m \in \mathbb{N}$ nombre de magasins

$d \in \mathbb{N}$ 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

$$\forall 1 \leq i \leq f \forall 1 \leq j \leq m \forall 1 \leq k \leq d,$$

$D_{i,j,k}$ est la quantité de fluide i demandée au magasin j lors de la demande k

Fonction objectif

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

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 $\text{D2} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

1	Problem:	PbMagasin					
2	Rows:	11					
3	Columns:	12					
4	Non-zeros:	36					
5	Status:	OPTIMAL					
6	Objective:	CoutTotal = 9.5	(MINimum)				
7							
8	No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
9							
10	1	RespectStock[F1,M1]					
11		NU		2.5		2.5	-1
12	2	RespectStock[F1,M2]					
13		B		0.5		1	
14	3	RespectStock[F1,M3]					
15		B		0		2	
16	4	RespectStock[F2,M1]					
17		NU		1		1	-2
18	5	RespectStock[F2,M2]					
19		B		1		2	
20	6	RespectStock[F2,M3]					
21		NU		1		1	-1
22	7	RespectDemande[F1,D1]					
23		NS		2	2	=	2
24	8	RespectDemande[F1,D2]					
25		NS		1	1	=	2
26	9	RespectDemande[F2,D1]					
27		B		0	-0	=	
28	10	RespectDemande[F2,D2]					
29		NS		3	3	=	3
30	11	CoutTotal	B	9.5			
31							
32	No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
33							
34	1	D[F1,M1,D1]	B	2	0		
35	2	D[F1,M1,D2]	B	0.5	0		
36	3	D[F1,M2,D1]	NL	0	0		< eps
37	4	D[F1,M2,D2]	B	0.5	0		
38	5	D[F1,M3,D1]	NL	0	0		1

```

39      6 D[F1,M3,D2]  NL          0          0          1
40      7 D[F2,M1,D1]  NL          0          0          3
41      8 D[F2,M1,D2]  B           1          0
42      9 D[F2,M2,D1]  NL          0          0          3
43     10 D[F2,M2,D2]  B           1          0
44     11 D[F2,M3,D1]  NL          0          0          3
45     12 D[F2,M3,D2]  B           1          0
46
47 Karush-Kuhn-Tucker optimality conditions:
48
49 KKT.PE: max.abs.err = 0.00e+00 on row 0
50         max.rel.err = 0.00e+00 on row 0
51         High quality
52
53 KKT.PB: max.abs.err = 0.00e+00 on row 0
54         max.rel.err = 0.00e+00 on row 0
55         High quality
56
57 KKT.DE: max.abs.err = 0.00e+00 on column 0
58         max.rel.err = 0.00e+00 on column 0
59         High quality
60
61 KKT.DB: max.abs.err = 0.00e+00 on row 0
62         max.rel.err = 0.00e+00 on row 0
63         High quality
64
65 End of output

```

1.2.2 Cas particulier 1.2

1.2.3 Cas particulier 2