ENSEEIHT

Recherche operationnelle

2022-2023

Rapport du TP1

Yasmine Charifi et Oussakel Rachida

Contents

List of Figures

1 Assemblage

1.1 Probleme PL

L'objectif de ce problème est de permettre à l'usine de savoir comment répartir le travail entre les deux modèles de voitures pour que la marge totale soit la plus grande possible.

• Les ensembles sur lesquels on travaille :
Les voitures : Cars.

• La variable qu'on veut minimiser :
- v[i](Matrice d'entier) avec indice i: parcourt l'ensemble des Cars.

• Les paramètres qu'on ne peut pas maîtriser mais dont on connaît la valeur:
- benefparVoiture[i in Cars] : bénéfice par voiture
- Surface [i in Cars] : surface occupé par chaque type de voiture
- Heure [i in Cars] : heure de fabrication de chaque voiture

• Le choix des contraintes
Nous avons ainsi choisit de modéliser ce problème en utilisant à priori trois contraintes qui limitent les choix des valeurs de décision :
- Une première contrainte qui s'assure du respect d'heure du travail.
- Une deuxième contrainte qui s'assure du respect de la surface du parking.
- Une troisième qui s'assure du respect de la quantité limite de la fabrication des voitures luxieuses.

Nous avons ainsi appliqué le modèle décrit sur les données du sujet. Ce qui nous génère le résultat suivant où la fonction objective prend une valeur de :

```
Problem:    ModelAssemblage
Rows:       4
Columns:    2 (2 integer, 0 binary)
Non-zeros:  7
Status:     INTEGER OPTIMAL
Objective:  BeneficeTotal = 10284000 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 RespectHeureDeTravail
                                  60                          60
     2 RespectSurface
                               14970                       15000
     3 RespectDemandeL
                                 645                         800
     4 BeneficeTotal
                          1.0284e+07

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 v[S]         *            426             0
     2 v[L]         *            645             0
```

Figure 1: Fonction objective pour un problème PL

Ainsi, la combinaison optimale consiste à produire 426 voitures standards et 645 luxueuses.

1.2 Probleme PLNE

La problèmatique permettra à l'usine de savoir comment répartir le travail de fabrication de lot (100 voitures) entre les deux modèles pour maximiser la marge de bénéfice.
Ainsi, la variable cherchée sera l'ensemble de lot à produire par voiture.

Nous avons ainsi appliqué le modèle décrit sur les données du sujet. Ce qui nous génère le résultat suivant où la fonction objective prend une valeur de :

```
Problem:
Rows:       3
Columns:    2 (2 integer, 0 binary)
Non-zeros:  5
Status:     INTEGER OPTIMAL
Objective:  BeneficeTotal = 980000 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 RespectHeureDeTravail
                                  58                          60
     2 RespectSurface
                               12000                       15000
     3 RespectDemandeL
                                   8                           8

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 v(S)         *              2             0
     2 v(L)         *              8             0
```

Figure 2: Fonction objective pour un problème PLNE

Dans ce cas de figure, l'usine générera plutot 2 lots de voitures standards et 8 lots de voitures luxueuses. Ainsi, on remarque une chutte de bénéfice, car dans ce cas, l'usine n'atteindra pas la limite des heures de travail (58< 60). Chose qui rend le problème PL plus efficace que ce dernier.

2 Gestion du personnel

L'objectif de ce problème est de trouver l'affectation minimisant le coût total de formation associés au couple personne travail sachant qu'on a N personnes et N travails.

• Les ensembles sur lesquels on travaille :
- Personnel : les personnes pouvant réaliser les formations;
- Competences : les compétences permettant la réalisation des travaux par le personnel ;

• La variable qu'on veut minimiser :
- var x[i,j] avec
indice i: parcourt l'ensemble du Personnel.
indice j : parcourt l'ensemble des Competences.

• Les paramètres qu'on ne peut pas maîtriser mais dont on connaît la valeur:
— c[i in Personnel, j in Competences] : le cout de chaque formtion de chaque personne i ayant une competence j.

• Le choix des contraintes Nous avons ainsi choisit de modéliser ce problème en utilisant à priori deux contraintes qui limitent les choix des valeurs de décision :
— Une première contrainte qui s'assure du fait que chaque travail est realisé par une seule personne .
— Une deuxième contrainte qui s'assure du fait que chaque personne doit être affectée à un travail et un seul.
Nous vons ainsi appliqué notre modèle sur l'ensemble de données suivant :

```
data;

set Personnel :=
p1
p2;

set Competences:=
c1
c2;

param c : c1 c2:=
p1 30 40
p2 34 50;
```

Figure 3: Exemple de données

ce qui donne un cout minimal de 75 :

```
Problem:
Rows:       4
Columns:    4
Non-zeros:  8
Status:     OPTIMAL
Objective:  Cout = 74 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 TacheassocieeUnePersonne(c1)
                    NS             1             1             =            34
     2 TacheassocieeUnePersonne(c2)
                    NS             1             1             =            50
     3 PersonneRealiseuneTache(p1)
                    NS             1             1             =           -10
     4 PersonneRealiseuneTache(p2)
                    B              1             1             =

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 x(p1,c1)     NL             0             0                           6
     2 x(p2,c1)     B              1             0
     3 x(p1,c2)     B              1             0
     4 x(p2,c2)     B              0             0
```

Figure 4: Fonction objective pour gestion du personnel

3 Applications en optimisation pour l'e-commerce

3.1 Premier cas

L'objectif de ce problème est de minimiser le cout de fluides demandés par les clients à partir du stock des magasins.

• Les ensembles sur lesquels on travaille :
- Les differents magasins produisants les fluides : Magasins.
- Les differents fluides proposés : Fluides.
- Les demandes des clients : Demandes.

• La variable qu'on veut minimiser :
- Q[i,j,k] avec
indice i: parcourt l'ensemble des fluides.
indice j: parcourt l'ensemble des magasins.
indice k: parcourt l'ensemble des demandes.

• Les paramètres qu'on ne peut pas maîtriser mais dont on connaît la valeur:
— Le coût unitaire par fluide pour les différents magasins: cout .
— Le stock de fluides par magasin : stock.
— Les demandes figurantes sur le tableau (a) : SDemandes.

La variable de décision dont on souhaite déterminer la valeur est la quantité par fluides qu'on peut prendre de chaque magasin pour satisfaire les demandes.

• Le choix des contraintes Nous avons ainsi choisit de modéliser ce problème en utilisant à priori deux contraintes qui limitent les choix des valeurs de décision :
— Une première contrainte qui s'assure que la somme des stocks pour un fluide dans l'ensemble des magasins est supérieure ou égale à la quantité demandée de ce fluide dans l'ensemble des demandes.
— Une deuxième contrainte qui s'assure que la somme des quantités prises pour un fluide dans l'ensemble des magasins est égale à la quantité demandée de ce fluide dans l'ensemble des demandes.

Nous avons ainsi appliqué le modèle décrit sur les données du sujet. Ce qui nous génère le résultat suivant où la fonction objective prend une valeur de 9.5 :

Ecommerce1.sol.txt

```
Problem:
Rows:       8
Columns:    12
Non-zeros:  24
Status:     OPTIMAL
Objective:  Cout = 9.5 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 stockdisponible(F1,M1)
                    NU           2.5                         2.5            -1
     2 stockdisponible(F1,M2)
                    B            0.5                           1
     3 stockdisponible(F1,M3)
                    B              0                           2
     4 stockdisponible(F2,M1)
                    NU             1                           1            -2
     5 stockdisponible(F2,M2)
                    B              1                           2
     6 stockdisponible(F2,M3)
                    NU             1                           1            -1
     7 QuantiteVerifiee(F1)
                    NS            -3            -3             =            -2
     8 QuantiteVerifiee(F2)
                    NS            -3            -3             =            -3
```

Figure 5: Fonction objective pour le premier cas

3.2 Deuxième cas

Dans ce cas, l'objectif est de minimiser le cout de produit (vendu par lot) demandés par les clients à partir du stock des magasins. Ainsi, on se place dans la perspective PLNE. Ainsi qu'on ne traite plus l'ensemble des demandes comme une seule mais on les déscritise, et c'est la raison pour laquelle nous avons ajouté la troisième contrainte qui s'assure que la quantité de produit demandé par un client est disponible en stock.

Nous appliquons ainsi le modèle sur les memes données que précedemment. Ce qui donne le résultat suivant où la fonction objective prend une valeur de 10 :

Ecommerce2.sol.txt

```
Problem:
Rows:       22
Columns:    12 (12 integer, 0 binary)
Non-zeros:  36
Status:     INTEGER OPTIMAL
Objective:  Cout = 10 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 stockdisponiblePouruneDemande(P1,M1,D1)
                                   1                         2.5
     2 stockdisponiblePouruneDemande(P1,M1,D2)
                                   1                         2.5
     3 stockdisponiblePouruneDemande(P1,M2,D1)
                                   1                           1
     4 stockdisponiblePouruneDemande(P1,M2,D2)
                                   0                           1
     5 stockdisponiblePouruneDemande(P1,M3,D1)
                                   0                           2
     6 stockdisponiblePouruneDemande(P1,M3,D2)
                                   0                           2
     7 stockdisponiblePouruneDemande(P2,M1,D1)
                                   0                           1
     8 stockdisponiblePouruneDemande(P2,M1,D2)
                                   1                           1
```

Figure 6: Fonction objective pour le deuxième cas

3.3 Troisième cas

L'objectif de ce problème est de minimiser le cout de fluides demandés par les clients à partir du stock des magasins en ajoutant des frais de livraison(fixe et variable).

• Les variables qu'on veut minimiser :
- Q[i,j,k] avec
indice i: parcourt l'ensemble des fluides.
indice j: parcourt l'ensemble des magasins.
indice k: parcourt l'ensemble des demandes.

- FraisExpedition[d,m] : avec
indice d: parcourt l'ensemble des demandes.
indice m: parcourt l'ensemble des magasins.
cette variable est déclarée comme binaire, ainsi on lui affecte 1 si le magasin M livre un fluide demandé par un client et 0 sinon.

• Les paramètres qu'on ne peut pas maîtriser mais dont on connaît la valeur:
— Le coût unitaire par fluide pour les différents magasins: cout .
— Le stock de fluides par magasin : stock.
— Les demandes figurantes sur le tableau (a) : SDemandes.
— Les frais d'expedition fixes figurantes sur le tableau (d) : expeditionFixe.
— Les frais d'expedition variables figurantes sur le tableau (e) : expeditionVariable.

• Le choix des contraintes :
Nous avons ainsi choisit de modéliser ce problème en ajoutant à priori deux contraintes qui limitent les choix des valeurs de décision :
— Une première contrainte qui s'assure que s'il n'y a pas de demande dans un magasin m, la valeur de FraisExpediton associée à la demande sera nulle.
— Une deuxième contrainte qui s'assure que s'il y a une demande dans un magasin m, la valeur de FraisExpediton associée à la demande sera non nulle.

Nous avons ainsi appliqué le modèle décrit sur les données du sujet. Ce qui nous génère le résultat suivant où la fonction objective prend une valeur de 368 :

```
Problem:    ModelEcommerceCas3
Rows:       23
Columns:    18 (18 integer, 6 binary)
Non-zeros:  78
Status:     INTEGER OPTIMAL
Objective:  Cout = 368 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 stockdisponible[P1,M1]
                                   1                         2.5
```

Figure 7: Fonction objective pour le troisième cas

3.4 Quatrième cas

Dans ce cas de figure,chaque magasin décide d'embaucher un livreur pour assurer toutes les livraisons qui le concerne. Ce dernier peut donc partir du magasin avec l'ensemble des colis et les livrer aux différents clients. Ainsi, l'objectif sera de minimiser la distance parcourue par chaque livreur . clients.

• L'ensembles sur lequel on travaille :
- Les differentes positions accessibles par un livreur : Positions.
- Les clients sauf le magasin : Clients.

• La variable qu'on veut minimiser : - D[p1,p2] : avec
indice p1,p2: parcourt l'ensemble des positions parcourues par le livreur.
- Directionsc in Clients L'ordre dans lequel on va parcourir les positions.
cette variable est déclarée comme binaire, ainsi on lui affecte 1 si le livreur passe de p1 à p2 et 0 sinon.

• Les paramètres qu'on ne peut pas maîtriser mais dont on connaît la valeur:
— La matrice des distances figurantes sur le tableau (f) : distances.
— Le nombre de clients à qui on compte livrer les demandes : nbClients.

• Le choix des contraintes :
Nous avons ainsi choisit de modéliser ce problème en définissant à priori deux contraintes qui limitent les choix des valeurs de décision :

— La première contrainte s'assure qu'on a livré toutes les demandes au bon nombre de client.
- Les deux contraintes qui suivent permettent d'assurer qu'un seul client est livré lors d'un passage.
— La quatrième contrainte s'assure que notre modèle ne choisit pas la distance nulle ,autrement dit, le livreur ne reste pas à la même position.
— La cinquième contrainte s'assure que le livreur ne retourne pas au magasin tant qu'il n'a pas fini la livraison de la totalité des demandes.
— Les dernières contraintes permettent de borner chaque direction.

Nous avons ainsi appliqué le modèle décrit sur les données du sujet. Ce qui nous génère le résultat suivant où la fonction objective prend une valeur de 22 :

Cas4Ecommerce.sol.txt

```
Problem:    ModelEcommerceCas4
Rows:       54
Columns:    41 (41 integer, 36 binary)
Non-zeros:  183
Status:     INTEGER OPTIMAL
Objective:  distance = 22 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 unSeulClientEstLivre1[Alpha]
                                   1             1             =
     2 unSeulClientEstLivre1[c1]
                                   1             1             =
     3 unSeulClientEstLivre1[c2]
```

Figure 8: Fonction objective pour le quatrième cas