الجمهـــوريـــة الجـزائـــريـــة الديــــمقراطيـــة الشــعبية
République Algérienne Démocratique et Populaire
وزارة الــتــعــلــيـــم الــعـــالـــــي و الــبــحــث الــعــلــمــي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
كلية العلوم الدقيقة والتطبيقية
Faculté des sciences exactes et appliquées

Département d'informatique

L3 INFORMATIQUE

MINI PROEJET

Modelisation d'un Problème d'horaire de personnel, d'un Problème de flot à coût minimum et la résolution des deux problèmes avec le solveur CPLEX

Eudiants :
REKKAB Nesrine Hanane
FORTAS Abdeldjalil Mohamed Mokhtar

Module :
Programmation Linéaire

8 décembre 2016

Exercice 1 : Problème d'horaire de personnel

Nous souhaitons établir un horaire quotidien, sachant que chaque jour est divisé en périodes et en supposant que nous avons pu estimer un nombre minimum d'employés devant être affecté durant chaque période. Chaque jour est divisé en quarts de travail de 8 heures. Plusieurs quarts partagent une même période, mais chaque quart exige un salaire particulier. Nous souhaitons savoir combien d'employés doit-on affecter à chaque quart de travail de façon à minimiser le total des salaires versés, en respectant le nombre minimum d'employés pour chaque période.
Les données du problème sont données dans la Table 1.

		Jour (j)					
	Périodes	Quart 1	Quart 2	Quart 3	Quart 4	Quart 5	**Minimum employés**
Jour (j)	6–8	X					48
	8–10	X	X				79
	10–12	X	X				65
	12–14	X	X	X			87
	14–16		X	X			64
	16–18			X	X		73
	18–20			X	X		82
	20–22				X		43
	22–24				X	X	52
	00–06					X	15
	Salaires	170	160	175	180	195	

Table 1: données du problème d'horaire de personnel

Eclaircissement du problème :

1. Chaque jour est divisé en périodes
2. L'estimation d'un nombre minimal d'employés affectés durent chaque période
3. Chaque jour est dévisé en quarts de travail de 8 heures
4. Les quarts (plusieurs) partagents la même période
5. Pour chaque quart $\rightarrow$ un salaire

Explication du modèle :
On pose : x_j nombre d'employés affectés au quart j (j appartenant à $\{1,2,3,4,5\}$)
On cherche à minimiser la fonction objectif :
F = $170x_1 + 160x_2 + 175x_3 + 180x_4 + 195x_5$

Pour chaque période, le nombre d'employés affectés aux différents quarts doit couvrir le minimum d'employés requis pour cette période.

Soit le PL :

$$F = 170x_1 + 160x_2 + 175x_3 + 180x_4 + 195x_5$$

$$\begin{aligned} x_1 &>= 48 \\ x_1 + x_2 &>= 79 \\ x_1 + x_2 &>= 65 \\ x_1 + x_2 + x_3 &>= 87 \\ x_3 + x_4 &>= 64 \\ x_3 + x_4 &>= 73 \\ x_3 + x_4 &>= 82 \\ x_4 &>= 43 \\ x_4 + x_5 &>= 52 \\ x_5 &>= 15 \end{aligned}$$

$$x_j >= 0 \quad \text{j = 1,2,3,4,5}$$

Problème de redondance :
$x_1 + x_2 >= 79 \rightarrow x_1 + x_2 >= 65$
Cette contrainte est redondante qu'on peut supprimer.

De même pour cette conrainte : $x_3 + x_4 >= 82 \rightarrow x_3 + x_4 >= 73$

Les contraintes de signes ne peuvent pas être supprimées.

Le PL final :

$$F = 170x_1 + 160x_2 + 175x_3 + 180x_4 + 195x_5$$

$$\begin{aligned} x_1 &>= 48 \\ x_1 + x_2 &>= 65 \\ x_1 + x_2 + x_3 &>= 87 \\ x_3 + x_4 &>= 64 \\ x_3 + x_4 &>= 73 \\ x_4 &>= 43 \\ x_4 + x_5 &>= 52 \\ x_5 &>= 15 \end{aligned}$$

$$x_j >= 0 \quad \text{j = 1,2,3,4,5}$$

Le modèle AMPL :

```
var x1 >= 0;
var x2 >= 0;
var x3 >= 0;
var x4 >= 0;
var x5 >= 0;

minimize fct_objectif: 170*x1 + 160*x2 + 175*x3 + 180*x4 + 195*x5;

subject to testing1 : x1 >= 48;
subject to testing2 : x1 + x2 >= 79;
subject to testing4 : x1 + x2 + x3 >= 87;
subject to testing5 : x2 + x3 >= 64;
subject to testing6 : x3 + x4 >= 82;
subject to testing8 : x4 >= 43;
subject to testing9 : x4 + x5 >= 52;
subject to testing0 : x5 >=15;
```

La résolution : la Solution optimale (obtenue par CPLEX) :
$(x_1 , x_2 , x_3 , x_4 , x_5) =$ (48, 31, 39, 43, 15)

```
ampl: option solver './cplex';
ampl: model exercice1.mod;
ampl: solve
ampl? ;
CPLEX 12.6.3.0: optimal solution; objective 30610
0 dual simplex iterations (0 in phase I)
ampl: display x1,x2,x3,x4,x5,fct_objectif;
x1 = 48
x2 = 31
x3 = 39
x4 = 43
x5 = 15
fct_objectif = 30610

ampl:
```

Problème : le nombre d'employés doit toujours être entier, donc l'hypothèse de divisibilité n'est pas satisfaite dans le modèle.

Exercice 2 : Problème de flot à coût minimum

Réseau de distribution Considérons deux usines (U1 et U2), un centre de distribution (CD), et deux entrepôts (E1 , E2). Chaque usine fabrique un certain nombre d'unités d'un même produit (l'offre). Chaque entrepôt requiert un certain nombre d'unités de ce même produit (la demande). Sur chaque lien (arc) du réseau, il y a un coût de transport par unité de produit (coût unitaire). De plus, sur certains arcs, il y a une capacité maximale sur le nombre d'unités transportées. Le réseau considéré est représenté dans la Figure 1. L'objectif est de minimiser le coût de transport total.

Pour modéliser ce problème, nous devrions, dans un premier temps, identifier les variables intrinsèques. Nous désignerons par x i,j le nombre d'unités du produit transportées sur l'arc (i, j) (i.e. entre les sommets i et j).

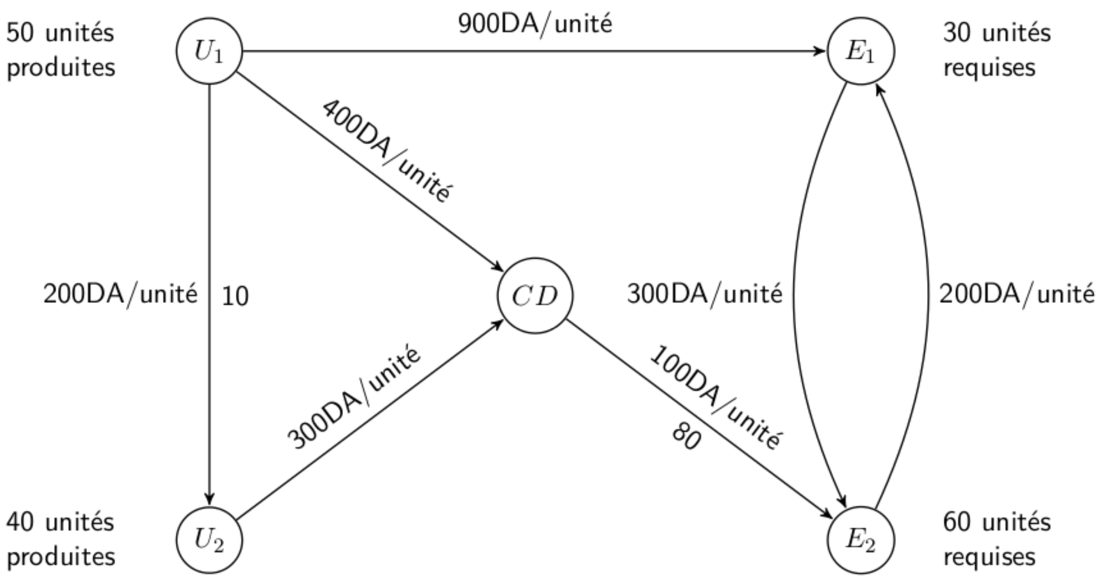

Figure 1: Réseau de distribution

Eclaircissement du problème :

1. Deux usines (U1,U2)
2. Un centre de distribution (CD)
3. Deux entrepôts (E1,E2)
4. Chaque usine manufacture un certain nombre d'unités d'un même produit (offre)
5. Chaque entrepôt requiert un certain nombre d'unités de ce même produit (demande)
6. Sur chaque lien (arc) du réseau, il y a un coût de transport par unité de produit (coût unitaire)
7. Sur certains arcs, il y a une capacité sur le nombre d'unités transportées
8. Objectif : minimiser le coût de transport total

Explication du modèle :

1. $X_{i,j}$: nombre d'unités du produit transportées sur l'arc (i,j) (entre les sommets i et j)
2. Fonction Objectif :
   min F = $2X_{U1,U2} + 4X_{U1,CD} + 9X_{U1,E1} + 3X_{U2,CD} + X_{CD,E2} + 3X_{E1,E2} + 2X_{E2,E1}$
3. Conservation du flot : en chaque sommet du réseau,flot sortant − flot entrant =nombre d'unités produites (usine) -nombre d'unités requises (entrepôt) 0 (CD)
4. Capacité (sur certains arcs) Exemple, pour l'arc (U1 ,U2) : $X_{U1,U2}$ 10
5. Contraintes de positivité

Le PL final :

$$\min F = 2X_{U1,U2} + 4X_{U1,CD} + 9X_{U1,E1} + 3X_{U2,CD} + X_{CD,E2} + 3X_{E1,E2} + 2X_{E2,E1}$$

$$\begin{gathered}
X_{U1,U2} + X_{U1,CD} + X_{U1,E1} = 50 \\
-X_{U1,U2} + X_{U2,CD} = 40 \\
-X_{U1,CD} - X_{U2,CD} + X_{CD,E2} = 0 \\
X_{E1,E2} - X_{E2,E1} = -30 \\
-X_{U1,U2} - X_{CD,E2} - X_{E1,E2} + X_{E2,E1} = -60 \\
X_{U1,U2} <= 10 \\
X_{CD,E2} <= 80
\end{gathered}$$

$$X_{U1,U2} >= 0 , X_{U1,CD} >= 0 , X_{U1,E1} >= 0 , X_{U2,CD} >= 0 , X_{CD,E2} >= 0 , X_{E1,E2} >= 0 , X_{E2,E1} >= 0$$

Le modèle AMPL :

```
var u1u2 >= 0;
var u1cd >= 0;
var u1e1 >= 0;
var u2cd >= 0;
var cde2 >= 0;
var e1e2 >= 0;
var e2e1 >= 0;
var e3>=0;
var e4>=0;

minimize fct_objectif: 2*u1u2+4*u1cd+9*u1e1+3*u2cd+cde2+2*e2e1;

subject to testing1 : u1u2 + u1cd + u1e1 = 50;
subject to testing2 : -1 * u1u2 + u2cd = 40;
subject to testing3 : -1 * u1cd - u2cd + cde2 = 0;
subject to testing4 : -1 * u1e1 + e1e2 - e2e1 =-30;
subject to testing5 : -1 * cde2 - e1e2 + e2e1 =-60 ;
subject to testing6 : u1u2 +e3= 10;
subject to testing7 : cde2 +e4= 80;
```

La résolution : la Solution optimale (obtenue par CPLEX) :
$(X_{U1,CD}\ ,\ X_{U1,E1}\ ,\ X_{U2,CD}\ ,\ X_{CD,E2}\ ,\ X_{E1,E2}\ ,\ X_{E2,E1}) = (0{,}40{,}10{,}40{,}80{,}0{,}20)$

```
fcmam5:~/Desktop/ample/scd$ ./ampl
ampl: option solver './cplex';
ampl: model exercice2.mod;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 490
3 dual simplex iterations (0 in phase I)
ampl: display u1u2,u1cd,u1e1,u2cd,cde2,e1e2,e2e1,e3,e4, fct_objectif;
u1u2 = 0
u1cd = 40
u1e1 = 10
u2cd = 40
cde2 = 80
e1e2 = 0
e2e1 = 20
e3 = 10
e4 = 0
fct_objectif = 490

ampl:
```

Problème : Le nombre d'unités transportées doit toujours être une valeur entière, donc l'hypothèse de divisibilité n'est pas satisfaite dans ce modèle. Dans ce cas particulier, la solution est entière En fait, pour tout problème de flot à coût minimum (avec paramètres à valeurs entières), il existe toujours une solution optimale entière.

Comment resoudre un PL avec AMPL et le Solveur CPLEX

1. Nous devons charger le solveur CPLEX
2. Nous chargeons son fichier qui se trouvait dans le répertoire courant en exécutant : option solver './cplex' ;
3. Nous chargeons notre modèle en tapant : model nom-fichier.mod ;
4. Nous appelons le solveur avec la commande : solve ;
5. Enfin pour l'affichage du résultat : display suivie des noms des variable séparés pas des , et finir toujours avec ;
6. Pour le minimum d'une fonction : display :30610.