Travaux Pratiques:

Optimisation (IBM ILOG Optimization Studio)

Professeur : HADJ ALI Allel

Étudiant : LAM Minh Triet

Date : 12/02/2021

Table des matières

I. INTRODUCTION :

Dans ce projet, nous utilisons l' IBM ILOG Optimization Studio pour optimiser les solutions aux problèmes.

Il y a deux façons de gérer ce programme : CPLEX ou programme d'interface IBM ILOG.

II. EXAMPLE 1:

1. CPLEX

Dans ce projet, nous devons résoudre un exemple pour comprendre le fonctionnement de CPLEX et l'interface d'optimisation d'IBM en résolvant un problème linéaire.

Max x1 + 2 x2
Sous certaines conditions
x1 + 3x2 <= 21
-x1 + 3x2 <= 18
x1 - x2 <= 5
x1, x2 >= 0

```
\ENCODING=ISO-8859-1
\Problem name: myprop1

Maximize
 obj1: x1 + 2 x2
Subject To
 c1: x1 + 3 x2 <= 21
 c2: -x1 + 3 x2 <= 18
 c3: x1 - x2 <= 5
 c4: x1 >= 0
 c5: x2 >= 0
End
```

Tout d'abord, nous utilisons le CPLEX pour résoudre ce problème.

```
CPLEX> read myprop1.lp
Problem 'myprop1.lp' read.
Read time = 0.00 sec. (0.00 ticks)
CPLEX> optimize
Version identifier: 20.1.0.0 | 2020-11-10 | 9bedb6d68
Tried aggregator 1 time.
LP Presolve eliminated 2 rows and 0 columns.
Reduced LP has 3 rows, 2 columns, and 6 nonzeros.
Presolve time = 0.00 sec. (0.00 ticks)

Iteration log . . .
Iteration:     1   Dual infeasibility =            0.000000
Iteration:     2   Dual objective     =           17.000000

Dual simplex - Optimal:  Objective =  1.7000000000e+01
Solution time =    0.00 sec.  Iterations = 2 (1)
Deterministic time = 0.01 ticks  (6.69 ticks/sec)

CPLEX> display solution objective
Dual simplex - Optimal:  Objective =  1.7000000000e+01
CPLEX> display solution variables -
Variable Name           Solution Value
x1                            9.000000
x2                            4.000000
```

“Display solution dual -” montrera toute la solution des variables. Nous avons ajouté quelques variables de relâchement à une contrainte d'inégalité pour la transformer en une égalité.

```
CPLEX> display solution dual -
Constraint Name            Dual Price
c1                           0.750000
c3                           0.250000
All other dual prices in the range 1-5 are 0.
CPLEX> display solution slack -
Constraint Name           Slack Value
slack c2                    15.000000
slack c4                    -9.000000
slack c5                    -4.000000
All other slacks in the range 1-5 are 0.
```

Pour effectuer une analyse de sensibilité, l'Optimiseur interactif prend en charge <<objectif>>. CPLEX affiche les plages suivantes pour l'analyse de sensibilité de la fonction objectif.

CPLEX affiche chaque variable, son coût réduit et la fourchette dans laquelle son coefficient de fonction objectif peut varier sans forcer un changement de la base optimale. La valeur actuelle de chaque coefficient de fonction objectif est également affichée à titre de référence. L'analyse de sensibilité de la fonction objectif est utile pour analyser la sensibilité de la solution optimale au coût ou au profit associé à chaque variable.

CPLEX peut également afficher des plages de sensibilité à la limite inférieure avec la commande :

```
CPLEX> display sensitivity objective -
                                            OBJ Sensitivity Ranges
Variable Name       Reduced Cost          Down          Current            Up
x1                          zero        0.6667           1.0000     +infinity
x2                          zero       -1.0000           2.0000        3.0000
CPLEX> display sensitivity lb -
                                          Lower Bound Sensitivity Ranges
Variable Name       Reduced Cost          Down          Current            Up
x1                          zero     -infinity             zero        9.0000
x2                          zero     -infinity             zero        4.0000
```

CPLEX peut également afficher les plages de sensibilité de la limite supérieure avec la commande :

```
CPLEX> display sensitivity ub -
                                          Upper Bound Sensitivity Ranges
Variable Name       Reduced Cost          Down          Current            Up
x1                          zero        9.0000        +infinity     +infinity
x2                          zero        4.0000        +infinity     +infinity
```

<<display sensitivity rhs ->> affiche les plages suivantes pour l'analyse de sensibilité du côté droit (RHS) :

```
CPLEX> display sensitivity rhs -
                                            RHS Sensitivity Ranges
Constraint Name       Dual Price          Down          Current            Up
c1                        0.7500        5.0000          21.0000       51.0000
c2                          zero        3.0000          18.0000     +infinity
c3                        0.2500       -5.0000           5.0000       21.0000
c4                          zero     -infinity             zero        9.0000
c5                          zero     -infinity             zero        4.0000
```

2. IBM ILOG:

Nous pouvons utiliser le ILOG d'IBM pour créer un programme optimisé pour notre problème. Tout d'abord, nous avons créé un nouveau projet :

pharma_float.mod | P2.mod

```
/*********************************************
 * OPL 20.1.0.0 Model
 * Author: lamm
 * Creation Date: 26 fÃ©vr. 2021 at 15:08:26
 *********************************************/
dvar float x1;
dvar float x2;

maximize
  x1 + 2*x2;

subject to {
  x1 + 3*x2 <= 21;
  -x1 + 3*x2 <= 18;
  x1 - x2 <= 5;
  x1 >= 0;
  x2 >= 0;
}

```

Ensuite, nous créons une configuration et nous exécutons la solution :

Problem browser | Variables | Breakpoints

Solution with objective 17

Name	Value
Decision variables (2)	
x1	9
x2	4

Les outils nous aident dans de nombreux aspects de la solution, que nous pouvons choisir dans la barre d'outils :

Problems | Scripting log | Solutions | Conflicts | Relaxations

```
// solution (optimal) with objective 17
// Quality There are no bound infeasibilities.
// There are no reduced-cost infeasibilities.
// Maximum Ax-b residual              = 0
// Maximum c-B'pi residual            = 0
// Maximum |x|                        = 9
// Maximum |slack|                    = 15
// Maximum |pi|                       = 0.75
// Maximum |red-cost|                 = 0
// Condition number of unscaled basis = 9.3e+00
//

x1 = 9;
x2 = 4;
```

III. EXEMPLE 2 :

Dans ce problème, nous allons optimiser un ensemble de données. Il est utile d'utiliser les fichiers .dat pour construire de nombreuses données pour un même fichier modèle.

Dans un premier temps, nous allons construire un fichier modèle général.

```
/*********************************************
 * OPL 20.1.0.0 Model
 * Author: lamm
 * Creation Date: 26 fÃ©vr. 2021 at 15:15:48
 *********************************************/
//lire dans un fichier le nb d'objet et le poids max
int nbObjet = ...;
int poidsMax = ...;
//dÃ©clarer un intervalle d'entiers de 1 Ã  nbObjet
range objets = 1..nbObjet;
//dÃ©clarer des tableaux indexÃ©s sur les objets,
//ils seront remplis en lisant le fichier de donnÃ©es
int poids[objets] = ...;
int valeur[objets] = ...;
//dÃ©clarer les variables de dÃ©cisions
dvar boolean x[objets];
//modele
maximize
  sum (i in objets) valeur[i]*x[i];
  subject to {
    sum(i in objets)
      poids[i]*x[i] <= poidsMax;
  }
```

Ensuite, nous créons un problème spécifique suite à la construction du fichier modèle.

```
/*********************************************
 * OPL 20.1.0.0 Data
 * Author: lamm
 * Creation Date: 26 fÃ©vr. 2021 at 15:21:12
 *********************************************/
nbObjet = 6;
poidsMax = 15;

poids = [1,2,3,4,5,6];
valeur= [4,5,6,7,8,9];
```

Lorsque nous exécutons, le logiciel comprendra l'entrée du fichier de données :

Problem browser | Variables | Breakpoints

Solution with objective 30

Name	Value
Data (5)	
poidsMax	15
valeur	[4 5 6 7 8 9]
poids	[1 2 3 4 5 6]
nbObjet	6
objets	1..6
Decision variables (1)	
x	[1 1 1 1 1 0]

Finalement, on a cette solution :

Problems Scripting log Solutions Conflicts Relaxations Engine log **Statistics** Profil

```
// solution (optimal) with objective 30
// Quality Incumbent solution:
// MILP objective                              3.0000000000e+01
// MILP solution norm |x| (Total, Max)         5.00000e+00  1.00000e+00
// MILP solution error (Ax=b) (Total, Max)     0.00000e+00  0.00000e+00
// MILP x bound error (Total, Max)             0.00000e+00  0.00000e+00
// MILP x integrality error (Total, Max)       0.00000e+00  0.00000e+00
// MILP slack bound error (Total, Max)         0.00000e+00  0.00000e+00
//

x = [1
         1 1 1 1 0];
```

IV. TRAVAIL À RENDRE :

Une entreprise fabrique deux produits A, B. Ces produits utilisent 3 molécules différentes, qu'on note ici H, P, et M.

Après calculation des quantités de matières premières nécessaires par boîte, ce problèmes peut être modélisé comme suit :

```
/*********************************************
 * OPL 20.1.0.0 Model
 * Author: lamm
 * Creation Date: 26 fÃ©vr. 2021 at 15:35:06
 *********************************************/
dvar int x1;
dvar int x2;

maximize
  6.5*x1 + 11.5*x2;

subject to {
  2.5*x1 + 7.5*x2 <= 2404;
  0.125*x1 + 0.125*x2 <= 51;
  17.5*x1 + 10*x2 <= 5950;
  x1 >= 0;
  x2 >= 0;
}
```

1. Le plan de production optimal :

Quand les variables est un nombre entier, on a la solution :

```
// solution (optimal) with objective 4032
// Quality Incumbent solution:
// MILP objective                              4.0320000000e+03
// MILP solution norm |x| (Total, Max)         4.08000e+02  2.76000e+02
// MILP solution error (Ax=b) (Total, Max)     0.00000e+00  0.00000e+00
// MILP x bound error (Total, Max)             0.00000e+00  0.00000e+00
// MILP x integrality error (Total, Max)       0.00000e+00  0.00000e+00
// MILP slack bound error (Total, Max)         0.00000e+00  0.00000e+00
//

x1 = 132;
x2 = 276;
```

2. Le plan de production optimal dans l'hypothèse où x1 et x2 sont réels :

Nous faisons passer les variables de l'entier au flottant.

```
/*********************************************
 * OPL 20.1.0.0 Model
 * Author: lamm
 * Creation Date: 26 fÃ©vr. 2021 at 15:35:06
 *********************************************/
dvar float x1;
dvar float x2;

maximize
  6.5*x1 + 11.5*x2;

subject to {
  2.5*x1 + 7.5*x2 <= 2404;
  0.125*x1 + 0.125*x2 <= 51;
  17.5*x1 + 10*x2 <= 5950;
  x1 >= 0;
  x2 >= 0;
}
```

Problems | Scripting log | Solutions | Conflicts | Relaxations | Engine log | Statistics | Pr

```
// solution (optimal) with objective 4036
// Quality There are no bound infeasibilities.
// There are no reduced-cost infeasibilities.
// Max. unscaled (scaled) Ax-b resid.          = 4.54747e-13 (5.68434e-14)
// Max. unscaled (scaled) c-B'pi resid.        = 0 (0)
// Max. unscaled (scaled) |x|                  = 276.8 (276.8)
// Max. unscaled (scaled) |slack|              = 886 (276.8)
// Max. unscaled (scaled) |pi|                 = 32 (8)
// Max. unscaled (scaled) |red-cost|           = 0 (0)
// Condition number of scaled basis            = 5.3e+00
//

x1 = 131.2;
x2 = 276.8;
```

3. Les coûts marginaux (prix fictifs) de chacune des ressources :

On suppose que x1 et x2 sont réels.

margin.mod

```
/*********************************************
 * OPL 20.1.0.0 Model
 * Author: lammi
 * Creation Date: Feb 27, 2021 at 12:12:11 AM
 *********************************************/
dvar float h;
dvar float p;
dvar float m;

minimize
  2404*h + 51*p + 5950*m;
  subject to {
    2.5*h + 0.125*p + 17.5*m >= 6.5;
    7.5*h + 0.125*p + 10*m >= 11.5;
    h>=0;
    p>=0;
    m>=0;
  }
```

Le coût marginal est la variation du coût total qui survient lorsque la quantité produite est augmentée d'une unité ; c'est-à-dire qu'il s'agit du coût de production d'une unité supplémentaire d'un bien. Nous appelons h, p, m selon 3 conditions de contrainte.

Après avoir exécuté le programme pour avoir la solution.

Problems Scripting log Solutions Conflicts Relaxations Engine log Statistics Prof

```
// solution (optimal) with objective 4036
// Quality There are no bound infeasibilities.
// There are no reduced-cost infeasibilities.
// Max. unscaled (scaled) Ax-b resid.          = 0 (0)
// Max. unscaled (scaled) c-B'pi resid.        = 4.54747e-13 (4.54747e-13)
// Max. unscaled (scaled) |x|                  = 32 (32)
// Max. unscaled (scaled) |slack|              = 32 (32)
// Max. unscaled (scaled) |pi|                 = 886 (2214.4)
// Max. unscaled (scaled) |red-cost|           = 0 (0)
// Condition number of scaled basis            = 7.7e+02
//

h = 1;
p = 32;
m = 0;
```

Les coûts marginaux sont donc :

- 1 pour H.
- 32 pour P.
- 0 pour M.

4. Les ressources critiques :

Nous pouvons voir que le coefficient de H et P est supérieur à 0, c'est pourquoi H et P sont les ressources critiques. De plus, P est la ressource la plus critique.

5. L'impact sur les bénéfices si le stock disponible de M est seulement de 5750g :

Tout d'abord, nous voyons que M n'est pas le coût marginal. Ce n'est donc peut-être pas le facteur le plus puissant pour affecter le bénéfice.

Ensuite, nous vérifions la solution ci-dessus, avec x1 = 131,2 et x2 = 276,8. Pour la molécules M:

17.5*131.2 + 10*276.8 = 5064.

Si M = 5750, le montant n'a pas d'incidence sur les produits. En outre, l'entreprise économise (5950 - 5750) = 200(g) de la molécule M.

6. L'analyse de sensibilité pour les deux coefficients de l'objectif et les trois coefficients des seconds membres des constraints :

Pour cette exigence, nous utilisons le CPLEX pour analyser la sensibilité.

```
CPLEX> enter
Enter name for problem: pharmapb
Enter new problem ['end' on a separate line terminates]:
Maximize
6.5x1 + 11.5x2
st
C1:
2.5x1 + 7.5x2 <= 2404
C2:
0.125x1 + 0.125x2 <= 51
C3:
17.5x1 + 10x2 <= 5950
C4:
x1 >= 0
C5:
x2 >= 0
end
CPLEX> display problem all
Maximize
 obj1: 6.5 x1 + 11.5 x2
Subject To
 C1: 2.5 x1 + 7.5 x2 <= 2404
 C2: 0.125 x1 + 0.125 x2 <= 51
 C3: 17.5 x1 + 10 x2 <= 5950
 C4: x1 >= 0
 C5: x2 >= 0
Bounds
 All variables are >= 0.
CPLEX> write pharmapb.lp
Problem written to file 'pharmapb.lp'.
CPLEX> |
```

On a l’analyse de sensibilité :

```
CPLEX> optimize
Version identifier: 20.1.0.0 | 2020-11-10 | 9bedb6d68
Tried aggregator 1 time.
LP Presolve eliminated 2 rows and 0 columns.
Reduced LP has 3 rows, 2 columns, and 6 nonzeros.
Presolve time = 0.02 sec. (0.00 ticks)

Iteration log . . .
Iteration:     1   Scaled dual infeas =             0.000000
Iteration:     2   Dual objective     =          4036.000000

Dual simplex - Optimal:  Objective =  4.0360000000e+03
Solution time =    0.03 sec.  Iterations = 2 (1)
Deterministic time = 0.01 ticks  (0.24 ticks/sec)

CPLEX> display sensitivity

Display Sensitivity Options:

lb                display a set of solution lower bound sensitivity ranges
objective         display a set of solution objective sensitivity ranges
rhs               display a set of solution right-hand side sensitivity ranges
ub                display a set of solution upper bound sensitivity ranges

Display which sensitivity analysis: objective
Display objective sensitivity for which variable(s): -
                                                OBJ Sensitivity Ranges
Variable Name      Reduced Cost            Down          Current              Up
x1                         zero          3.8333           6.5000         11.5000
x2                         zero          6.5000          11.5000         19.5000
CPLEX> display sensitivity lb -
                                              Lower Bound Sensitivity Ranges
Variable Name      Reduced Cost            Down          Current              Up
x1                         zero       -infinity             zero        131.2000
x2                         zero       -infinity             zero        276.8000
CPLEX> display sensitivity ub -
                                              Upper Bound Sensitivity Ranges
Variable Name      Reduced Cost            Down          Current              Up
x1                         zero        131.2000        +infinity       +infinity
x2                         zero        276.8000        +infinity       +infinity
CPLEX> display sensitivity rhs -
                                                RHS Sensitivity Ranges
Constraint Name      Dual Price            Down          Current              Up
C1                       1.0000       1813.3333        2404.0000       3060.0000
C2                      32.0000         40.0667          51.0000         56.2118
C3                         zero       5064.0000        5950.0000       +infinity
C4                         zero       -infinity             zero        131.2000
C5                         zero       -infinity             zero        276.8000
```

6. Parmi les trois molécules, Il serait le plus intéressant d'avoir 1g supplémentaire :

Comme le résultat ci-dessus, la molécule P est le plus intéressant d'avoir 1g supplémentaire, parce que son coût marginal est le plus élevé des trois molécules (P = 32).

7. Le nouveau plan de production optimal :

La compagnie pharmaceutique décide d'acheter jusqu'à 250g de H en plus, chez un autre fabricant, qui vend le gramme 20 centimes de plus que leur producteur habituel. (En particulier, le bénéfice sur le produit A n'est plus que de 6.5 - 2.5*0.2 = 6 euros, et le bénéfice sur le produit B n'est plus que de 11.5 - 7.5*0.2 = 10 euros, si on utilise ces nouvelles molécules). Montrer que ce nouveau problème peut être modélisé comme suit :

```
/*********************************************
 * OPL 20.1.0.0 Model
 * Author: lammi
 * Creation Date: Feb 27, 2021 at 12:59:22 AM
 *********************************************/
dvar float x1;
dvar float x2;
dvar float x3;
dvar float x4;

maximize
  6.5*x1 + 11.5*x2 + 6*x3 + 10*x4;

subject to {
  2.5*x1 + 7.5*x2 <= 2404;
  2.5*x3 + 7.5*x4 <= 250;
  0.125*x1 + 0.125*x2 + 0.125*x3 + 0.125*x4 <= 51;
  17.5*x1 + 10*x2 + 17.5*x3 + 10*x4 <= 5950;
  x1 >= 0;
  x2 >= 0;
  x3 >= 0;
  x4 >= 0;
}
```

Ensuite, nous avons la solution :

```
// solution (optimal) with objective 4236
// Quality There are no bound infeasibilities.
// Max. unscaled (scaled) reduced-cost infeas. = 2.22045e-16 (2.22045e-16)
// Max. unscaled (scaled) Ax-b resid.          = 3.41061e-13 (2.13163e-14)
// Max. unscaled (scaled) c-B'pi resid.        = 8.88178e-16 (8.88178e-16)
// Max. unscaled (scaled) |x|                  = 293.467 (293.467)
// Max. unscaled (scaled) |slack|              = 1261 (293.467)
// Max. unscaled (scaled) |pi|                 = 32 (8)
// Max. unscaled (scaled) |red-cost|           = 0 (0)
// Condition number of scaled basis            = 1.5e+01
//

x1 = 81.2;
x2 = 293.47;
x3 = 0;
x4 = 33.333;
```

V. CONCLUSION :

Dans ce projet, nous pouvons voir que CPLEX et IBM ILOG sont les meilleurs outils pour résoudre le problème optimisé en suivant les instructions du professeur.

Ici, nous voyons la modification lorsque nous changeons les différents facteurs.