

Stéphane Canu

Janvier 2017, ASI, INSA Rouen

Le but du TP est d'identifier, de reformuler et de résoudre des programmes linéaires en nombres entiers à l'aide de logiciels standard parmi les plus efficaces (comme par exemple Cplex).

**Ex. 1 — programmation linéaire en nombres entiers**

- On cherche à résoudre le problème suivant

$$\left\{ \begin{array}{l} \min_{x_1, x_2 \in \mathbb{N}^2} \quad 3x_1 + 2x_2 \\ \text{avec} \quad 4x_1 + x_2 \leq 15 \\ \quad \quad x_1 + 2x_2 \leq 8 \\ \quad \quad x_1 + x_2 \leq 5 \end{array} \right.$$

- Écrire un programme CVX permettant de résoudre la relaxation continue du problème

```
cvx_begin
variables x1 x2 ;
maximize( 3*x1 + 2*x2 );
subject to
4*x1+ 2*x2 <= 15;
x1+ 2*x2 <= 8;
x1+x2 <=5;
x1 >= 0;
x2 >= 0;
cvx_end
```

Que pensez vous de la solution trouvée et du cout associé

- Écrire deux programmes CVX permettant de résoudre la relaxation continue du problème en ajoutant les contraintes  $x_1 \leq 2$  et  $x_1 \geq 3$

```
cvx_begin
variables x1 x2 ;
maximize( 3*x1 + 2*x2 );
subject to
4*x1+ 2*x2 <= 15;
x1+ 2*x2 <= 8;
x1+x2 <=5;
x1 >= 0;
x2 >= 0;
x1 >= 3;
cvx_end
```

```
cvx_begin
variables x1 x2 ;
maximize( 3*x1 + 2*x2 );
subject to
4*x1+ 2*x2 <= 15;
x1+ 2*x2 <= 8;
x1+x2 <=5;
x1 >= 0;
x2 >= 0;
x1 <= 2;
cvx_end
```

- d) Écrire un programme CVX permettant de résoudre la relaxation continue du problème sous forme matricielle standard

```
c = -[3 ; 2];
A = [4 2; 1 2; 1 1];
b = [15; 8 ; 5];

cvx_begin
    variables x(2) ;
    minimize( c'*x);
    subject to
        A*x <= b;
        x >= 0;
cvx_end
```

Vérifiez que votre formulation est bien correcte

- e) que fait la commande

```
help cplexmilp
```

- f) Résoudre le problème en utilisant `cplexmilp`

```
lb = 0*c;
ub = [];

ctype = ['I' 'I'];
options = cplexoptimset('Display', 'on', 'MaxTime', 10);

[x_ci, fval, exitflag, output] = cplexmilp(c, A, b, [], [], [], [], [], lb, ub, ctype, [], options);
```

- g) A quoi servent la variable `ctype` et la fonction `cplexoptimset` ?
2. On s'intéresse maintenant au problème du sac à dos

$$\begin{cases} \max_{z \in \{0,1\}^4} & 45z_1 + 30z_2 + 45z_3 + 10z_4 \\ & 3z_1 + 5z_2 + 9z_3 + 5z_4 \leq 16 \end{cases}$$

- a) Donnez une relaxation continue de ce problème sous une forme standard

```
c = -[45 30 45 10];
w = [3 5 9 5];

cvx_begin
    variables x(4) ;
    minimize ( c*x );
    subject to
        w*x <= 16 ;
        x >= 0 ;
        x <= 1 ;
cvx_end
```

- b) Résoudre le problème en utilisant `cplexmilp`

```
A = w;
b = 16;

lb = 0*c;
ub = [];

ctype = repmat('B', 1, 4);
options = cplexoptimset('Display', 'on', 'MaxTime', 10);

[x_ci, fval, exitflag, output] = cplexmilp(c, A, b, [], [], [], [], [], lb, ub, ctype, [], options);
```

- c) Proposez une stratégie de type « branch and bound » pour résoudre ce problème. Vérifiez que vos résultats sont bien corrects.

3. Proposez un programme utilisant `cplexmilp` permettant de résoudre le problème suivant :

$$\left\{ \begin{array}{l} \max_{x \in \mathbf{N}^3} \quad 20x_1 + 10x_2 + 10x_3 \\ \text{avec} \quad 2x_1 + 20x_2 + 4x_3 \leq 15 \\ \quad \quad 6x_1 + 20x_2 + 4x_3 = 20 \end{array} \right.$$