

Fiche de TP n°2

Linear programming

L'Objectif de ce travail est se familiariser avec l'utilisation de Minizinc (www.minizinc.org) pour résoudre des problèmes d'optimisation linéaires.

IMPORTANT : Sur l'interface graphique, choisir : “solver configuration” → COIN-BC.

1. Exécuter le code suivant :

```
var float: x;
var float: y;
var float: f=150*x+450*y;
constraint x>=0 /\ y>=0 /\ x<=120 /\ y<=70 /\ x+y <=140 /\ x+2*y<=180;
solve maximize f;
```

2. Maintenant, remplacer la ligne « `var float: f=150*x+450*y;` » par :

```
var float: f; constraint f=150*x+450*y;
```

Conclure.

3. Remplacer les lignes convenables par :

```
constraint f=160*x+300*y;
constraint x>=0 /\ y>=0 /\ x<=120 /\ y<=70 /\ x+y <=140 /\ 2*x+3*y<=181;
```

Exécuter le script, puis changer le type de toutes les variables “float” vers “int”

Comparer les résultats, que remarquer vous ?

4. Découvrir la syntaxe Minizinc, notamment : (voir MiniZinc_Tutorial.pdf)

Les types, Déclarations, paramètres, variable de décision, “variable’s domain”, “Relational Operators”, array, set, “List and Set Comprehensions”, data file « .dzn ».

La partie « output » peut être considéré comme un sous-programme, elle est formée essentiellement de chaînes de caractères. Elle peut contenir de boucles et d'instruction if-then-else-endif.

Exemple : ajouter la ligne suivante : `output ["x=\(x);\ny=\(y);"];`

Ensuite vous la remplacez par : `output ["x=\(x);"+["\ny=\(y);"]];`

5. Exécuter le script suivant :

```
set of int: v_rnd={3, 6,1, 8 ,4, 9, 0, 5, 7, 2};
array[1..10] of var v_rnd: v;
constraint forall(e in v_rnd)(exists(i in 1..10)(v[i]=e));
constraint forall(i,j in 1..10 where i<j )(v[i] <= v[j]);
solve satisfy;
```

Commenter brièvement chaque ligne.

Ajouter les instructions nécessaires pour afficher le tableau v sous les deux formes suivantes :

`v={0, 1, 2, 3, 4, 5, 6, 7, 8, 9};`

Les indices pairs dans la première ligne et les impairs sur la deuxième :

`1, 3, 5, 7, 9,`

`0, 2, 4, 6, 8,`

6. Faire une petite recherche (e.g. Wiki) sur : Constrained programming, MIP/CP Solver, Declarative programming subparadigms.

Puis donner une conclusion générale