

TP n°3

L'objectif de ce TP est de mettre en œuvre le solveur Minizinc pour gérer des problèmes d'affectation rencontrés dans l'organisation de conférences. La section suivante introduit des nouvelles fonctionnalités du solveur.

Minizinc

La fonction `bool2int` transforme un booléen en entier (1 pour *true* et 0 pour *false*). Elle permet en particulier de *réifier* une contrainte: `bool2int` appliquée à une contrainte donne un entier représentant sa valeur de vérité. Dans l'exemple ci-dessous, la contrainte assure pour tout $i = 1, \dots, M$ que le nombre d'occurrences de i dans le tableau x est au plus p .

```
int: p = ...;
array[1..N] of var 1..M: x;
constraint forall(i in 1..M) (sum(j in 1..N) (bool2int(x[j] = i) <= p));
```

Minizinc est capable de résoudre des problèmes d'optimisation. Ainsi, il est possible de minimiser ou de maximiser une fonction objectif, ce qui apparaît dans l'instruction `solve`. L'exemple ci-après montre la minimisation de la fonction linéaire $\sum_i c_i x_i$.

```
int: N = ...;
array[1..N] of 1..N: c = ...;
array[1..N] of var 1..N: x;
solve minimize sum(i in 1..N) (c[i]*x[i]);
```

Il est possible également de déclarer une variable de coût de la manière suivante.

```
var int: cost = sum(i in 1..N) (c[i]*x[i]);
solve minimize cost;
```

Il est souvent utile de manipuler des *tableaux* de données ou de variables en deux dimensions. Dans l'exemple ci-dessous, `tab` est une matrice d'entiers de N lignes et M colonnes. L'accès à la donnée sur la ligne i et la colonne j s'écrit `tab[i,j]`.

```
int: N = 3;
int: M = 2;
array[1..N, 1..M] of int: tab = array2d(1..N, 1..M, [ 3, 1,
                                                         2, 8,
                                                         7, 5 ] );
```

Si les lignes n'ont pas toutes la même dimension alors il est préférable de définir un tableau à une dimension dont chaque élément est un ensemble, de la manière suivante.

```
int: N = 3;
array[1..N] of set of int: tab = [ {1, 3},
                                   {2, 4, 7, 9},
                                   {} ];
```

Planification d'une conférence

Le problème est de construire le planning d'une conférence. Cette conférence est composée de 11 sessions (chaque session accueillant des conférenciers) et on suppose que les sessions ont toutes une même durée. Le planning est représenté comme une séquence de créneaux, un créneau pouvant accueillir au plus 3 sessions en parallèle. De plus, il faut prendre en compte les contraintes suivantes.

1. La session 4 a lieu avant la session 11.
2. La session 5 a lieu avant la session 10.
3. La session 6 a lieu avant la session 11.
4. La session 1 ne doit pas être en parallèle des sessions 2, 3, 5, 7, 8, 10.
5. La session 2 ne doit pas être en parallèle des sessions 3, 4, 7, 8, 9, 11.
6. La session 3 ne doit pas être en parallèle des sessions 5, 6, 8.
7. La session 4 ne doit pas être en parallèle des sessions 6, 8, 10.
8. La session 6 ne doit pas être en parallèle des sessions 7, 10.
9. La session 7 ne doit pas être en parallèle des sessions 8, 9.
10. La session 8 ne doit pas être en parallèle de la session 10.

Enfin, on veut minimiser le nombre de créneaux nécessaires. Modéliser et résoudre ce problème en séparant le modèle (fichier **mzn**) et les données (fichier **dzn**).

Gestion des conférenciers

Le problème est d'allouer N créneaux à N conférenciers en prenant en compte leurs préférences. Les préférences sont données dans la table ci-dessous. Par exemple, le conférencier 5 peut intervenir sur le créneau 3 avec la préférence 2 et le créneau 4 avec la préférence 1 (il préfère donc le créneau 3).

Conférencier	Créneaux avec préférences
1	(3, 3), (4, 1), (5, 2), (6, 4)
2	(3, 1), (4, 2)
3	(2, 1), (3, 2), (4, 4), (5, 3)
4	(2, 3), (3, 2), (4, 1)
5	(3, 2), (4, 1)
6	(1, 4), (2, 2), (3, 1), (4, 3), (5, 6), (6, 5)

Le problème est d'affecter les créneaux aux conférenciers de manière à optimiser une fonction objectif, par exemple maximiser la somme totale des préférences ou maximiser la préférence minimale.

Suite

La suite du TP peut être consacrée aux problèmes précédents en utilisant de nouveaux jeux de données ou au traitement de problèmes vus en TD.