

## Devoir 2 - Modélisation en programmation par contraintes

Remise le 9 novembre (avant minuit) sur Moodle pour tous les groupes.

### Consignes

- Le devoir doit être fait par groupe de 2 au maximum. Il est recommandé d'être 2.
- Lors de votre soumission sur Moodle, donnez vos 5 modèles minizinc (.mzn) à la racine d'un seul dossier compressé (matricule1\_matricule2\_Devoir2.zip).
- Indiquez vos noms et matricules en commentaires au dessus des fichiers .mzn soumis.
- Veillez à commenter votre code. Il n'y a aucun rapport à remettre pour ce devoir.
- Assurez vous que vos réponses s'impriment. Les fonctions d'output ont été générés pour vous. Vous n'avez pas à les modifier.
- Toutes les consignes générales du cours (interdiction de plagiat, etc.) s'appliquent pour ce devoir.

### Logiciels requis

Si ce n'est pas déjà fait, vous aurez besoin de télécharger [MiniZinc et son environnement de développement](#).

### Conseils

Ce devoir est volontairement *challenging* pour obtenir la totalité des points. Voici quelques conseils pour le mener à bien :

1. Prenez-y vous tôt. Il y a également le temps d'apprentissage de Minizinc à considérer.
2. Travaillez efficacement en équipe, et répartissez vous bien les tâches.
3. Tirez le meilleur parti des séances de laboratoire encadrées afin de demander des conseils.
4. Prenez avantage des contraintes globales offertes par la librairie de Minizinc.

Bonne chance!

### Enoncé

Pour ce deuxième devoir, il vous est demandé de modéliser différents problèmes, de difficulté croissante, sur MiniZinc. Un [tutoriel](#) se trouve à même la documentation, vous pourrez passer au travers avant d'attaquer les exercices. Pour chaque exercice, un fichier de base .mzn vous est fourni. Vous êtes fortement encouragés à utiliser l'API de minizinc et d'identifier les contraintes globales les plus pertinentes pour résoudre les problèmes. Vos modèles devraient être capables de résoudre chaque configuration en moins de 2 minutes avec le solveur Chuffed (utilisez *Time limit* dans la configuration solveur). Si vous modifiez les variables proposées, vous devez également modifier l'output pour la console. Vous aurez probablement à ajouter vos propres variables.

Lors de la soumission, envoyez seulement vos cinq modèles Minizinc. Ils seront exécutés et ils devront tourner sans erreur et imprimer une réponse. Les fonctions d'output ont été codés pour vous, vous n'avez donc pas à les modifier. Vous pouvez ajouter des commentaires si vous le jugez nécessaire.

## Problème 0 - introductory level (1 pt)

On vous demande de trouver un **nombre** composé de 4 **chiffres** qui satisfait les critères suivant :

- C'est un nombre pair.
- Le chiffre 0 n'est pas présent dans le nombre.
- Les 4 chiffres sont différents.
- Le chiffre à la position des milliers est supérieur à celui à la position des centaines.
- Le chiffre à la position des dizaines est inférieur à celui à la position des unités.
- Le chiffre à la position des centaines est supérieur à celui à la position des unités.
- La somme des 4 chiffres est supérieure à 15.
- Le produit des 3 derniers chiffres (chiffre à la position des centaines  $\times$  chiffre à la position des dizaines  $\times$  chiffre à la position des unités) doit être minimisé.

Implémentez un modèle pour résoudre ce problème.

## Problème 1 - easy level (3 pts)

Vous devez sélectionner un certain nombre d'attractions à construire dans votre ville afin d'attirer plus de touristes.

Vous devez sélectionner entre  $l$  et  $u$  attractions parmi un ensemble de  $A$  attractions qui vous est donné. Chaque attraction a un coût de construction associé.

On détermine l'attrait touristique de votre ville en sommant l'attrait touristique (positif ou négatif) de chaque paire différente d'attraction sélectionnée. La matrice d'attrait touristique ( $A \times A$ ) est fournie. Par exemple : l'attrait touristique du CAFÉ et du PARC est +4, l'attrait touristique du CAFÉ et du CINÉMA est +1, l'attrait touristique du PARC et du CINÉMA est -3. Si vous construisez le CAFÉ, le PARC et le CINÉMA, l'attrait touristique de la ville est 2 ( $4 + 1 - 3$ )

Vous devez sélectionner les attractions touristiques à construire afin d'avoir un attrait touristique total supérieur à  $m$  tout en minimisant le coût total.

Implémentez un modèle générique pour résoudre ce problème. Trois configurations vous sont données dans des fichiers .dzn. Votre modèle sera évalué sur ces configurations.

Afin de vous aider à vérifier votre modèle, on vous donne le coût optimal de l'instance 0 : 5

## Problème 2 - normal level (3 pts)

Vous êtes responsable de la construction de maisons dans un quartier résidentiel. Plusieurs pâtés de maisons sont vagues dans votre quartier et vous devez décider combien de maisons construire sur chacun de ces pâtés.

Le quartier est composé de  $h$  segments horizontaux et  $v$  segments verticaux. Attention, le quartier n'est pas nécessairement carré. Il y a donc  $h \times v$  pâtés de maisons sur lequel vous pouvez construire des maisons.

Pour chaque segment (horizontal et vertical) une demande minimale de maison à construire est fournie.

La Figure 1 représente un exemple avec 4 segments verticaux et horizontaux. Les chiffres dans les flèches bleues représentent la demande minimum par segment.

Si vous décidez de construire une ou plusieurs maisons sur un pâté de maison, vous devez payer le permis

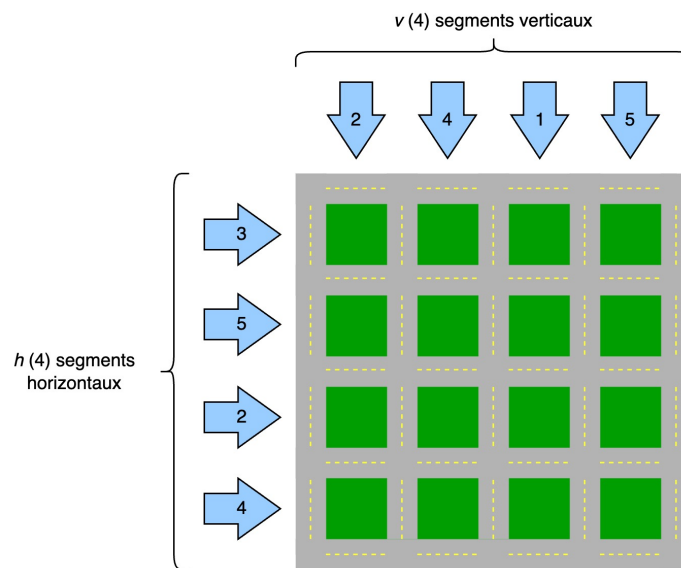


FIGURE 1 – Quartier Résidentiel

de construction associé (coût fixe). De plus, un prix de construction par maison est donné pour chaque pâté de maison (coût variable). Par exemple, si vous décidez de construire 2 maisons sur un pâté, vous devez payer le permis du pâté + 2 fois le coût d'une maison sur ce pâté. Vous ne pouvez pas construire plus de  $m$  maisons par pâté.

On vous demande de déterminer le nombre de maisons à construire sur chaque pâté afin de satisfaire la demande et minimiser les coûts de construction.

Implémentez un modèle générique pour résoudre ce problème. Trois configurations vous sont données dans des fichiers .dzn. Votre modèle sera évalué sur ces configurations.

Afin de vous aider à vérifier votre modèle, on vous donne le coût optimal de l'instance 0 : **40**

### Problème 3 - hard level (4 pts)

Vous êtes maintenant responsable de la construction de tours au centre ville. Comme dans le numéro précédent, le centre ville est représenté par des segments verticaux et horizontaux. Cette fois-ci la dimension est fixe : 5 segments horizontaux et 5 segments verticaux (25 pâtes où construire).

Vous devez construire des tours entre 1 et 5 étages sur chaque pâté.

Pour chaque segment horizontal et vertical du quartier, on ne peut pas avoir deux tours de la même taille.

De plus, le nombre de tours qu'un observateur situé aux alentours du quartier est capable de voir est fixé à une certaine valeur. Un observateur ne peut pas voir une tour si elle se trouve derrière une tour plus grande. La Figure 2 donne un exemple d'un champ de vision d'un observateur. Dans cet exemple, l'observateur ne voit que 3 tours (vert, rose, jaune). Les tours orange et bleue sont cachées par au moins une autre tour plus grande.

Le quartier est représenté par la Figure 3. Les nombres indiqués dans les flèches bleues représente le nombre de tours qu'un observateur situé à cet endroit est capable de voir.

Implémentez un modèle générique pour résoudre ce problème.

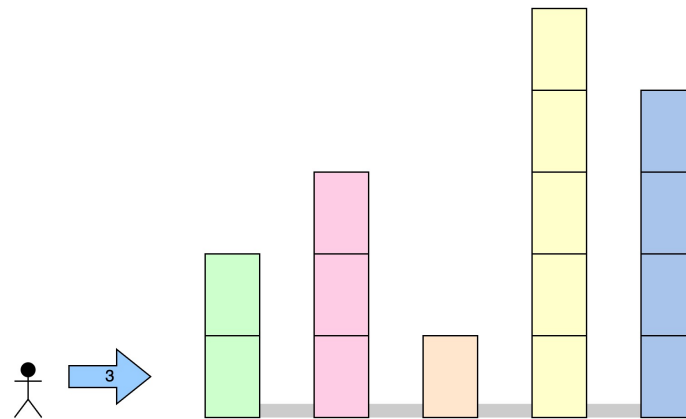


FIGURE 2 – Représentation du champ de vision

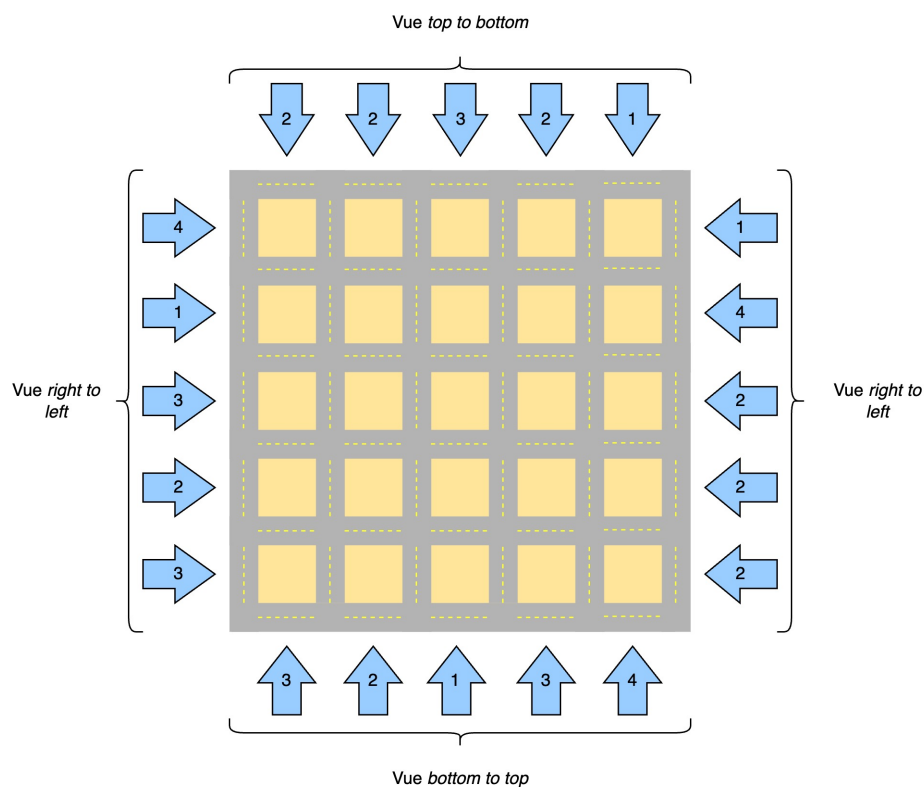


FIGURE 3 – Centre-Ville

### Problème 4 - extreme level (4 pts)

Vous êtes responsable de choisir  $P$  projets à accomplir en  $P$  jours. Vous devez faire l'horaire de vos 3 employés (ingénieur, architecte et électricien) afin de terminer les  $P$  projets.

Il y a  $N$  projets qui peuvent être réalisés ( $N \geq P$ ). Pour compléter un projet, l'ingénieur, l'architecte et l'électricien doivent chacun avoir travaillé une journée dessus. Vous devez donc assigner un projet à chaque employé à chaque jour. Les employés n'ont pas besoin de travailler immédiatement à la suite de l'autre sur un projet. Par exemple, l'ingénieur pourrait travailler sur le projet A lors de la première journée, l'électricien

pourrait travailler sur ce projet lors de la 4e journée et finalement l'architecte pourrait travailler sur ce projet lors de la 9e journée.

Il ne peut pas y avoir plus d'un employé assigné à un projet par jour.

Chaque projet complété rapporte un certain revenu. Par contre, il y a des frais de déplacement à considérer lorsqu'un employé passe d'un projet à un autre. Ces frais sont déterminés à l'aide d'une matrice  $P \times P$ . Attention, les frais de déplacement ne sont pas nécessairement symétriques (Le coût de passer du projet A au projet B n'est pas forcément le même que le coût de passer du projet B au projet A).

Certains projets doivent absolument être complétés. De plus, certains projets ont des contraintes sur l'ordre de passage des employés. Par exemple, le projet B pourrait demander à ce que l'architecte passe en premier, ensuite l'ingénieur et finalement l'électricien. Encore une fois, il peut y avoir plusieurs jours entre chaque passage.

On vous demande donc d'assigner à chaque employé, à chaque jour, un projet différent afin de maximiser vos profits.

Implémentez un modèle générique pour résoudre ce problème. Trois configurations vous sont données dans des fichiers .dzn. Votre modèle sera évalué sur ces configurations.

Afin de vous aider à vérifier votre modèle, on vous donne le coût optimal de l'instance 0 : **79**

On vous donne un exemple très simple (*4\_Extreme\_Data\_Exemple.dzn*) ainsi que la solution. Attention, les noms de variables ainsi que le format d'output ne sont pas exactement pareils que ceux des fichiers .mzn. Ceci est uniquement dans le but de faciliter la compréhension du problème.

```
enum PROJECT = {A, B, C, D}

P = 3

travel_costs = [0, 1, 1, 1
                1, 0, 1, 1
                1, 1, 0, 1
                1, 1, 1, 0]

set of PROJECT project_with_precedences = {B}
set of PROJECT project_must_be_done = {D}

precedences = [None, None, None
               Archi, Ing, Elec
               None, None, None
               None, None, None]

revenu = {5, 5, 5, 1}

=====
Solution
=====
Horaire :
    ARCHI    : B, D, A
    ING      : A, B, D
```

ELEC : D, A, B  
Revenus :  
 $5 + 5 + 1 = 11$   
Costs:  
Archi :  $[B,D] + [D,A] = 2$   
Ing :  $[A,B] + [B,D] = 2$   
Elec :  $[D,A] + [A,B] = 2$   
 $= 6$   
Profit :  
 $11 - 6 = 5$