

Devoir 2 - Modélisation en programmation par contraintes

Remise le 20 mars (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, indiquez la réponse retournée par votre code sous forme de commentaire en fin de fichier mzn.
- 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 encouragé 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 et indiquez la réponse retournée par votre code sous forme de commentaire en fin de fichier mzn (vos codes seront exécutés : ils doivent tourner sans erreur et retourner une solution identique ou équivalente à celle indiquée). Vous pouvez ajouter un commentaire de votre part après la solution si vous le jugez nécessaire.

1 Problème 1 - introductory level (1 pt)

Sauriez-vous trouver 4 entiers différents compris entre 10 et 100 dont la somme des carrés vaut le carré d'un autre entier le plus petit possible?

2 Problème 2 - easy level (3 pts)

Énigme des mariages : Rémy, David, Charles, Pierre et Édouard sont mariés à Émilie, Juliette, Maria, Élisabeth et Manon, mais on ne sait pas qui avec qui. Les couples s'appellent Dupont, Tremblay, Durand, Gagnon et Petit mais on ne sait pas de qui ils sont composés. Avec les indices suivants, déterminer qui appartient à quel couple, et depuis combien de temps chaque couple est marié (il n'y en a pas deux avec la même durée de mariage!).

- Juliette n'est pas mariée depuis aussi longtemps que Charles ou les Tremblays, mais depuis plus longtemps que David ou les Duponts.
- Élisabeth est mariée depuis deux fois plus longtemps que les Durands, mais deux fois moins longtemps que Rémy.
- Les Petits sont mariés depuis 10 ans de plus que Pierre et 10 ans de moins que Maria.
- David et Manon sont mariés depuis 25 ans de moins que les Gagnons qui, mariés depuis 30 ans, sont le couple marié depuis le plus longtemps.
- Ni Émilie ni les Duponts ont été mariés depuis le moins longtemps.
- Édouard est marié depuis 25 ans.

3 Problème 3 - normal level (3 pts)

Une entreprise doit faire réaliser N tâches sur T jours à S sous-traitants. Chaque tâche a été estimée à un coût et une durée différents par chaque sous-traitant, et on souhaite déterminer une attribution des tâches qui respecte l'horizon temporel fixé T . De plus, pour des questions de sécurité, le sous-traitant s'occupant de la tâche 1 et celui s'occupant de la tâche 2 ne peuvent s'occuper d'aucune autre tâche. Quel sera le coût minimum pour l'entreprise?

Si un sous-traitant hérite de plusieurs tâches, leur temps d'exécution se cumule. Un fichier `.dzn` vous permet de charger les données.

4 Problème 4 - hard level (3 pts)

Un grand tournoi très suivi de Quoridor à 2 joueurs est organisé et 13 champions sont inscrits. On demande de générer un calendrier de matchs nécessitant le moins de créneaux possible et faisant en sorte que chaque joueur rencontre chaque autre. Il y a trois tables de jeu pouvant accueillir une partie à chaque créneau, et on souhaite donc savoir quels joueurs jouent quel créneau et sur quelle table. On souhaite également que :

- chaque joueur ne joue pas trois créneaux de suite,
- le joueur 1 est un joueur vedette qui joue toujours à la table 1 et dont l'emploi du temps est déjà fixé pour des questions de programmation de retransmission télévisuelle sur QuoridorTV. Il joue contre le joueur 2, puis 3, puis 4, etc., et a un créneau de pause tous les deux matchs, comme exigé précédemment (sa première pause a donc lieu entre le joueur 3 et 4).

Formuler le problème comme un problème de minimisation.

5 Problème 5 - harder level (4 pts)

Ce grand tournoi est suivi d'un autre tournoi de Quoridor à 4 joueurs. Les 13 mêmes champions s'affrontent. Ce tournoi durera 14 jours, et chaque jour 2 matchs de 4 joueurs ont lieu. Cette fois-ci 2 sites de match sont dans 2 villes différentes, donc un joueur ne peut jouer qu'un seul des 2 matchs par jour et a le droit à au moins une journée de repos entre deux matchs si ceux-ci ne sont pas dans la même ville. Les organisateurs ont fixé des règles pour obtenir des résultats représentatifs pour chacun des joueurs, notamment :

- chaque joueur doit avoir rencontré chaque autre joueur en match,
- chaque joueur doit avoir fait entre 8 et 11 matchs,
- chaque joueur doit avoir joué au moins une fois sur chaque site.

Pouvez-vous générer un emploi du temps recevable indiquant quels joueurs participent aux différents matchs pour satisfaire ces contraintes?

Attention les matchs $[1, 2, 3, 4]$ et $[1, 2, 4, 3]$ sont équivalents, réfléchissez à une contrainte ou une modélisation qui permette au solveur de n'en considérer qu'un des deux pour une résolution plus efficace.