

Projet : Exportation de fromages

Remise le 10 décembre sur Moodle

Consignes

- Le projet doit être réalisé seul ou par binôme (de préférence par binôme).
- Soumettez un document (format pdf) reprenant votre rapport, ainsi que votre modèle MiniZinc (fichier `.mzn` et `.dzn`).
- Indiquez vos noms et matricules dans le titre des fichiers soumis.
- Veillez à rendre un rapport **structuré**, **clair** et **concis**, et si possible tapé. Les lacunes de forme seront pénalisées.
- Tout le matériel vu en cours peut être utilisé.
- 2 points seront attribués à la clarté des modèles et du rapport. Détaillez bien votre modèle et commentez vos contraintes MiniZinc.
- Même si vous n'arrivez pas à tout faire, expliquez votre démarche et ce qu'il vous manque (les démarches incomplètes peuvent être valorisées).
- L'utilisation d'une IA générative est **STRICTEMENT INTERDITE**.

Situation

Le Fromage du Monde, notre fromagerie des devoirs 1 et 2, a bien grandi et n'a jamais aussi bien porté son nom : en plus du Québec, elle exporte maintenant ses produits aux États-Unis. Ce sont donc 2 porte-conteneurs qu'elle doit remplir de ses variétés fromagères chaque semaine. Il est désormais possible de mélanger plusieurs fromages au sein d'un même conteneur. Ces derniers existent en 4 tailles pouvant contenir chacune 40, 60, 80 et 100 tonnes de fromage. Au moment de charger les porte-conteneurs, la fromagerie doit donc choisir la taille des conteneurs utilisés ainsi que les quantités de chaque fromage présent à l'intérieur. Le choix d'une taille et d'un mélange est appelé une configuration. Afin de simplifier sa logistique, la fromagerie décide d'utiliser au plus 4 configurations pour ses conteneurs tous pays confondus. Toutes les quantités de fromage (en tonnes) d'une configuration doivent être entières et les conteneurs doivent être remplis à capacité maximale. Ainsi, il est possible que certains pays reçoivent un surplus de fromage (*overstock*) ou que leur demande ne soit pas remplie (*understock*). Les quantités d'*overstock* tolérées pour chaque fromage sont les mêmes pour chaque pays et elles sont résumées dans la Table 1. La pénalité associée à chaque tonne d'*overstock* est de 1, tandis que celle associée à chaque tonne d'*understock*, bien plus problématique, est de 10. L'objectif de la fromagerie est d'avoir le moins de pénalités possible. Les demandes de chaque pays pour chaque variété de fromage sont résumées dans la Table 2. Enfin, on considère ici que la capacité des porte-conteneurs ne risque pas d'être excédée au vu des demandes des différents pays.

Munster	Comté	Épisses	Reblochon
10	20	20	0

TABLE 1 – Quantité d'*overstock* tolérée par chaque pays pour chaque variété de fromage (en tonnes)

	Munster	Comté	Éppoisses	Reblochon
Québec	40	20	30	50
États-Unis	20	30	40	50

TABLE 2 – Demande hebdomadaire de chaque pays pour chaque variété de fromage (en tonnes)

1 Modélisation et implémentation (15 pts)

1.1 Programmation par contrainte

1. Modélisez ce problème en utilisant la programmation par contraintes. Formalisez mathématiquement les données, les variables de décision, la fonction objectif ainsi que les contraintes. Souvenez-vous que la programmation par contraintes préfère ne pas utiliser de variables binaires, contrairement à la PLNE, limitez leur utilisation.
2. Implémentez votre modèle sous MiniZinc.
3. Construisez votre propre "mini" instance du problème (avec moins de données que l'original) afin de la résoudre à la main et de vérifier votre implémentation ; fournissez le `.dzn` correspondant. Expliquez pourquoi votre instance permet effectivement de tester la justesse de votre programme.
4. Résolvez le problème fourni avec MiniZinc. Produisez un fichier `.dzn` que vous rendrez également. Limitez le temps de résolution à 120 secondes et précisez le solveur utilisé ; trouvez-vous une solution optimale ? Détaillez la composition du porte-conteneurs envoyé vers les États-Unis.
5. Vous avez vu en cours qu'en programmation par contraintes, des contraintes redondantes peuvent permettre de supprimer des symétries et éventuellement améliorer le temps de résolution. Proposez une contrainte redondante à ajouter à votre modèle.

1.2 Programmation en nombres entiers

1. Modélisez ce problème en utilisant la programmation en nombres entiers. Formalisez mathématiquement les données, les variables de décision, la fonction objectif ainsi que les contraintes. Vous pouvez avoir au plus une contrainte non linéaire.
2. Implémentez votre modèle sous MiniZinc et vérifiez qu'il fonctionne sur votre "mini" instance.
3. Résolvez le problème fourni avec MiniZinc et les mêmes consignes de résolution que précédemment.
4. (Bonus +1) Proposez un modèle linéaire.

1.3 Analyse

Comparez vos deux modèles (nombre de variables, contraintes, solutions, temps de résolution, etc.).

2 Variante (5 pts)

1. La fromagerie a maintenant décidé de ne plus limiter le nombre de configurations qu'elle utilise pour ses conteneurs et de répondre exactement à la demande de chaque pays. Cependant, utiliser beaucoup de configurations différentes a un fort coût logistique, elle souhaite donc en avoir le

moins possible. Modélisez ce nouveau problème dans les deux paradigmes de programmation avec les mêmes consignes que précédemment.

2. Implémentez vos modèles sur MiniZinc et résolvez-les. Limitez le temps de résolution à 120 secondes et précisez le solveur utilisé ; trouvez-vous une solution optimale dans les deux cas ?
3. Comparez vos deux modèles (nombre de variables, contraintes, solutions, temps de résolution, etc.).