

Université de Paris - Master 1 Informatique - Programmation logique par contraintes

Examen du 8 janvier 2021 - Durée : 2 heures

Informations : Tous les documents non-électroniques sont autorisés. Le barème est donné à titre indicatif et peut être modifié.

Exercice 1 *Programmation puzzle* (4 points)

On considère le puzzle suivant:

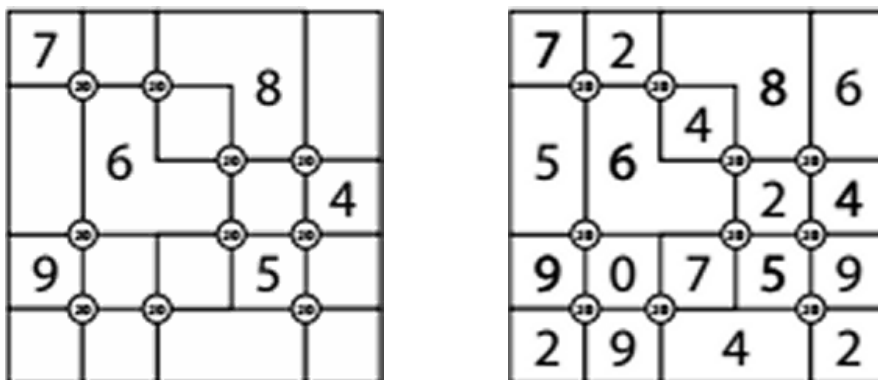


Figure 1: Une grille et sa solution

Chaque région de la grille doit être remplie par un nombre entre 0 et 9 de sorte que

- les nombres dans deux régions adjacentes (verticalement ou horizontalement) soient différents
- à chaque fois qu'il y a quatre régions qui se rencontrent en un point (indiqué par un petit rond), la somme de leurs nombres soient égale à 20.

Donnez un programme en ECLiPSeCLP qui résout la grille de l'exemple.

Exercice 2 *Simplex* (7 points)

On considère le problème suivant :

Minimiser $-Y - X$ par rapport à $X \geq 0, Y \geq 0$ et

$$\begin{aligned} 6 &\geq 3 * X - Y \\ X &\leq \frac{5}{2} \\ Y &\leq 2 \end{aligned}$$

- Visualisez le problème en dessinant un plan (axes: X et Y) avec les contraintes. Les contraintes définissent une zone dans le plan. Soyez précis.

- Appliquez l'algorithme simplexe (Il est simple d'obtenir une forme simplexe de base).
- Pour quelles valeurs de X et Y le minimum est-il atteint ?
- Donnez une requête en ECLiPSeCLP permettant d'obtenir le minimum.
- Donnez une fonction objective (à la place de $-Y - X$) à minimiser de sorte que le minimum soit atteint à plusieurs endroits.
- Est-ce que pour n'importe quelle fonction objective à minimiser le problème a toujours une solution ? Justifiez.
- Dans le problème original on impose en plus que X et Y doivent être des entiers. Quel est le minimum dans ce cas ? Donnez la réponse à partir du dessin (sans calcul).

Exercice 3 *Consistance* (5 points)

On considère la contrainte suivante sur les entiers : $X \geq 2 * Y \wedge 3 * X + 4 = 2 * Z \wedge Y + 12 \leq X + Z$ avec les domaines (intervalles) $X \in [2 \dots 8]$, $Y \in [1 \dots 8]$ et $Z \in [4 \dots 11]$.

- Est-ce que la contrainte est **nœud-consistante** ?
- Rendez la contrainte **arc-consistante**. Donnez uniquement les nouveaux domaines (qui ne sont plus forcément des intervalles).
- Rendez la contrainte **borne-consistante** en partant des domaines d'origine. Donnez les nouveaux domaines (qui sont des intervalles). Justifiez.
- Est-ce qu'en général l'arc-consistance d'une contrainte implique sa satisfaisabilité ?

Exercice 4 *Programmation* (4 points)

En utilisant la librairie `clpq` écrivez un prédicat `listecalcul` en ECLiPSeCLP qui étant donné une liste de taille au moins 3 de la forme `[0,_,_,...,_,_,_,10]` donne une liste, où chaque élément (à partir du troisième) de la liste est la moyenne de ses deux prédécesseurs.

Par exemple, `L=[0,_,_,_,_,10],listecalcul(L)` donne `L = [0,160/11,80/11,120/11,100/11,10]`.