

Université Paris Diderot - Sorbonne Paris Cité - Master 1 Informatique -
Programmation logique par contraintes
Examen du 16 janvier 2013 - Durée : 2 heures

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

Exercice 1 (5 points)

On considère le problème suivant :

Minimiser $-3X - 2Y$ par rapport à $X \geq 0, Y \geq 0$ et

$$\begin{aligned} X + Y &\leq 4 \\ 2X + Y &\leq 6 \end{aligned}$$

- Visualisez le problème en dessinant un plan (axes : X et Y) avec les contraintes.
- Appliquez l'algorithme simplex (Il est simple d'obtenir une forme simplex de base).
- On suppose qu'on n'est pas très sûr du coefficient -3 dans la fonction à minimiser ($-3X - 2Y$). On remplace donc -3 par $(-3 + \delta_1)$. Appliquez l'algorithme simplex en faisant les *mêmes* opérations (choix des variables et équations à pivoter) qu'en appliquant le simplex sur le problème original. Dans quelles conditions le résultat obtenu permet de conclure ce que c'est le minimum ?
- De la même façon on pourrait avoir une incertitude sur la valeur 4. On considère le problème : Minimiser $-3X - 2Y$ par rapport à $X \geq 0, Y \geq 0$ et

$$\begin{aligned} X + Y &\leq 4 + \delta_2 \\ 2X + Y &\leq 6 \end{aligned}$$

Appliquez l'algorithme simplex en faisant les *mêmes* opérations (choix des variables et équations à pivoter) qu'en appliquant le simplex sur le problème original. Dans quelles conditions le résultat obtenu permet de conclure ce que c'est le minimum ?

Exercice 2 (3 points)

On considère la contrainte $Z > X \wedge Y < X \wedge Z > Y$ avec les domaines (intervalles) $D(X) = [1..3], D(Y) = [0..2], D(Z) = [2..4]$.

- Indiquez, indépendamment, si cette contrainte est nœud-consistante, arc-consistante, borne-consistante, chemin-consistante.
- Si elle n'est pas nœud-consistante (resp. arc-consistante, borne-consistante, chemin-consistante), rendez la nœud-consistante (resp. arc-consistante, borne-consistante, chemin-consistante) sans éliminer des solutions évidemment. Justifiez.

Exercice 3 (3 points)

Un voyageur débarque aux pays des couleurs. À l'aéroport, il veut changer ses Euros en Coleuros (la monnaie du pays des couleurs). Au guichet on lui donne l'information suivante : Il y a quatre pièces de Coleuros : les vertes, jaunes, bleues et rouges. Deux pièces rouges plus trois pièces bleues valent ensemble 3 Euros 48 cents. Quatre pièces vertes plus une jaune plus une bleue valent 6 Euros 20 cents. Une pièce jaune et deux pièces vertes valent 3 Euros 76. Trois pièces vertes et une pièce jaune et une pièce bleue et une pièce rouge valent 6 Euros 04. Avec ces informations, le voyageur écrit le programme en GPROLOG suivant :

```
colorland(V,J,B,R) :-  
    2*R + 3*B #= 348,  
    4*V + J + B #= 620,  
    J + 2*V #= 376,  
    3*V + J + B + R #= 604.
```

En posant la question ?- colorland(V,J,B,R) GPROLOG répond :

```
B = _#24(0..116)
J = _#126(____.____)
R = _#43(0..174)
V = _#107(____.____)
```

- Donnez les 4 valeurs manquantes à la place des ___ de la sortie de GPROLOG. Justifiez **en détail** en donnant les calculs nécessaires (Plusieurs itérations sont nécessaires).
- Le voyageur souhaite connaître les valeurs exactes des pièces (pas seulement des intervalles). Comment doit-on modifier le programme pour cela ?

Exercice 4 (7 points)

On considère le jeu Bokkusu. Le but du jeu est de marquer en noir certaines cases de la grille (carré) donnée. Chaque case a deux valeurs : une valeur A et une valeur B. Les nombres à droite dénotent les valeurs A des cases de chaque ligne correspondante et les nombres en bas dénotent les valeurs B des cases de chaque colonne correspondante (Ces valeurs vont toujours de 1 à la taille de la grille en augmentant de 1 de gauche à droite ou du haut vers le bas). Les valeurs à gauche indiquent la somme des valeurs B des cases noires pour chaque ligne et les valeurs en haut indiquent la somme des valeurs A des cases noires pour chaque colonne. Un exemple d'une grille et sa solution est donné dans la figure 1. Dans la solution de cette exemple on a par exemple $7 = 1 + 2 + 4$ (les cases avec valeur B de 1, 2 et 4 sont marquées noir) et $6 = 2 + 4$ (les cases avec valeurs A de 2 et 4 sont marquées).

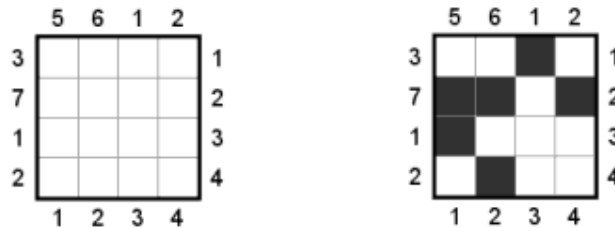


FIGURE 1 – Une grille Bokkusu et sa solution

- Écrivez un programme en GPROLOG qui résout la grille de l'exemple.
- Écrivez un programme en GPROLOG qui résout une grille quelconque (la taille et les valeurs à gauche et en haut sont des paramètres).

Exercice 5 (4 points)

L'entreprise Tetraonics doit embaucher des travailleurs intérimaires pendant 5 jours pour faire face à un pic de demande passager. Chaque travailleur peut travailler soit 2 jours **consécutifs** soit 3 jours **consécutifs**. Au moins 10 travailleurs sont nécessaires les jours 1, 3 et 5, tandis qu'au moins 15 travailleurs sont nécessaires les jours 2 et 4. Un travailleur qui travaille 2 jours consécutifs est payé 125 Euros par jour tandis qu'un travailleur qui travaille 3 jours consécutifs est payé 100 Euros par jour. Un travailleur qui travaille 2 jours ne peut pas commencer le jour 5 et un travailleur qui travaille 3 jours ne peut pas commencer les jours 4 et 5. Tetraonics veut minimiser le coût.

- Modélisez le problème comme un CSP. Il vous est demandé de **modéliser** et non pas de résoudre le problème.
- A cause d'un nombre limité de personnels de supervision pas plus de 10 travailleurs peuvent commencer chaque jour. Modifiez votre CSP en conséquence.
- Les réglementations de travail requièrent qu'au moins la moitié de l'argent soit dépensée pour les travailleurs qui travaillent 3 jours. Modifiez votre CSP en conséquence.
- Il y a quatre travailleurs qui veulent travailler les jours 1, 2 et 5. Modifiez votre CSP en conséquence.
- Donnez la requête qui permet à **YAP Prolog** de résoudre le problème.