

TP2 – Programmation linéaire en nombres entiers

Exercice 1 : résolution d'un PL en nombres entiers

Le script `exercice_1` est censé résoudre le programme linéaire en nombres entiers P'_3 selon la méthode de séparation/évaluation présentée en cours. Ce script appelle la fonction récursive `resolution`, que vous devez compléter. Cette fonction appelle la fonction `linprog` de Matlab, qui a déjà été vue lors du TP1. Les appels récursifs doivent construire un arbre de résolution « en profondeur d'abord ».

Lorsque vous obtiendrez la solution attendue, faites une copie de ce script, de nom `exercice_1_bis`, censé résoudre le programme linéaire P'_1 associé au programme linéaire relaxé P_1 du TP1. Comme x_1 et x_2 représentent des nombres de milliers de voitures, ces variables ne peuvent pas être des réels quelconques... Comparez le résultat avec la solution (x_1^*, x_2^*) du problème P_1 relaxé : aurait-il été judicieux de fixer la production hebdomadaire de voitures L et S en arrondissant, à l'aide de la fonction `floor`, les valeurs $1000 x_1^*$ et $1000 x_2^*$?

Exercice 2 : mise en échec du solveur de sudokus du TP1

Le solveur de sudokus du TP1 utilise la programmation linéaire à variables continues. Ce solveur fonctionne bien jusqu'au niveau de difficulté « expert ». Néanmoins, certains sudokus plus difficiles peuvent le mettre en échec. Par exemple, tentez de résoudre le sudoku `niveau9` avec le script `exercice_2` du TP1 :

```
>> exercice_2('niveau9')
```

Vous constatez que la valeur maximale de l'objectif est bien égale à $N^2 = 81$, mais que plusieurs cases ont gardé leur valeur initiale 0. Cela vient de ce que les variables de décision sont réelles et non pas booléennes.

Effectuez une copie du script `exercice_2`, de nom `exercice_2_bis`, que vous modifierez de façon à afficher, pour chaque case (i, j) , la valeur de la variable $x_{i,j,1}$. Vous constatez que certaines de ces valeurs ne sont pas entières, dans le cas du sudoku `niveau9`.

Exercice 3 : le sudoku vu comme un PL à variables booléennes

Il serait très compliqué de programmer la méthode de séparation/évaluation pour un PL en nombres entiers quelconque. La fonction `intlinprog` (*mixed-integer linear programming*) de Matlab est conçue pour résoudre efficacement ce type de PL par séparation/évaluation (cf. cours). En tapant `help intlinprog` dans la fenêtre Matlab, consultez la documentation de cette fonction.

Effectuez une nouvelle copie du script `exercice_2`, de nom `exercice_3`, que vous modifierez de manière à remplacer l'appel à `linprog` par un appel à `intlinprog`. Testez ce script sur le sudoku `tres_facile`. Lorsque vous obtiendrez le bon résultat, faites des tests sur les autres sudokus, par ordre de difficulté croissante : `facile`, `expert`, `niveau9`, `niveau10`, puis `diabolique`.

Vous pouvez assez facilement remplir un certain nombre de cases vides du sudoku `diabolique`. Si vous relancez le script `exercice_3` après modification du fichier `diabolique.txt`, vous observez que le nombre de nœuds de l'arbre de résolution varie fortement.