

Complexité et Récursivité

michel.vasquez@mines-ales.fr

février 2023

Pavage d'un polygone rectilinéaire avec des carrés

Le problème à résoudre consiste à recouvrir une surface plane avec un minimum de carrés. La zone à recouvrir est constituée de carrés unitaires : la plupart doivent être recouverts et d'autres non.

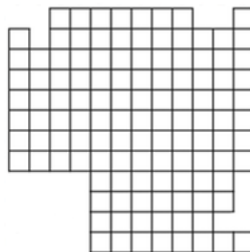
Ce sujet a été proposé par une compagnie d'imprimerie internationale :



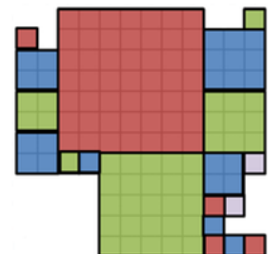
[ABOUT US](#) [OUR FOCUS](#) [OUR BRANDS](#)

THE CIMPRESS TECH CHALLENGE!

How **efficiently** can you
cover a grid like this...

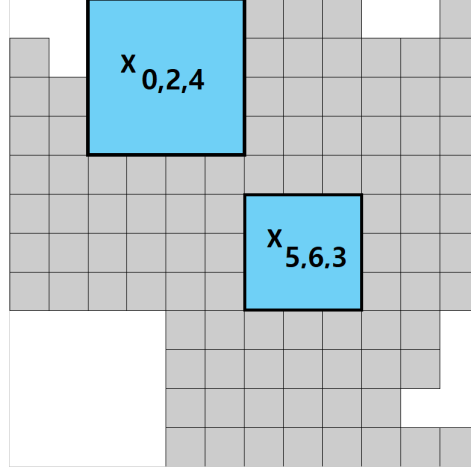


...with squares?



1 Modélisation

$X_{i,j,k} \Leftrightarrow$ carré de coin supérieur gauche (i, j) et de côté k . $X_{i,j,k}$ est valide s'il est à l'intérieur de la grille et s'il n'intersecte aucun trou qui altère cette grille.



Deux carrés $X_{i,j,k}$ valides : i indice de ligne, j indice de colonne.

Etant donné :

- \mathcal{G} : ensemble des cases unitaires de la grille *altérée*,
- $\mathcal{X} = \{X_{i,j,k}\}$: ensemble des carrés valides pour \mathcal{G}

Il s'agit de trouver un sous-ensemble $\{X_1 \cdots X_n\}$ d'éléments de \mathcal{X} tel que :

- $\mathcal{G} = X_1 \cup X_2 \cup \cdots \cup X_n$
- $X_i \cap X_j = \emptyset \quad 1 \leq i < j \leq n$
- n minimum

2 Formulation mathématique

Variables binaires :

- $x_{ijk} = 1$ si le carré $X_{i,j,k}$ fait partie de la solution,
- $x_{ijk} = 0$ sinon.

Partition minimale de la grille *altérée* :

$$\begin{aligned}
 & \text{Minimiser } \sum_{i,j,k} x_{ijk} \\
 & \forall (ii, jj) \in \mathcal{G} \quad \sum_{i \leq ii < i+k, j \leq jj < j+k} x_{ijk} = 1 \\
 & x_{ijk} \in \{0, 1\}
 \end{aligned}$$

3 Format des données

Une instance est définie par un fichier texte qui commence par la donnée de la *largeur* et de la *hauteur* du rectangle englobant la zone bidimensionnelle à couvrir. Ces deux valeurs sont suivies par une séquence de $largeur \times hauteur$ caractères : 0 si la case est libre (à recouvrir) 1 si c'est un trou (ou un obstacle, en tout cas une case que l'on ne doit pas couvrir). Par exemple, le fichier `s1.txt` contient les trois lignes de texte suivantes :

[illegible]

et représente puzzle suivant :

```
D:\vasquez\Squares>sqr 1
hauteur 6 largeur 8 nbt 5
. . . . .
x x x x x x x
x x x x x x x
x x x x x x x
x x . x x x x
x . x . x x x
x x x x . x x
```

où les '*' correspondent aux cases (carrés de taille 1 x 1) à recouvrir et les '.' aux défauts à ne pas recouvrir.

4 Travail demandé

Il s'agit de coder une fonction d'exploration exhaustive de l'espace des solutions pour trouver la couverture minimale d'une instance donnée. Vous disposez de 12 instances (`s1.txt` à `s12.txt`) pour tester votre programme. Pour chacune de celles-ci il faut indiquer : ses caractéristiques, le nombre minimum de carrés pour la recouvrir et le temps de calcul. Vous devrez remettre votre source code en C/C++ ou Python et un tableau de résultats du type :

Pb.	hight	width	#holes	#squares	cpu
sol1.txt	6	8	5	15	0.1
...	...				
sol12.txt	...				

Vous devrez également m'indiquer une estimation de la complexité de votre algorithme. Toutes ces informations doivent tenir dans **un seul fichier texte** de 3-4 pages. Ce document doit commencer par la liste (*Prénom-Nom,...*) des membres de l'équipe.