Rapport Projet RO203 - Jeux Unruly Singles

GOSSEC Yaël
VERRECHIA Thomas

22 avril 2023

Table des matières

1 Introduction **2**

2 Solveur utilisé **2**

3 Jeu Unruly **3**
3.1 Règles du jeu . . . 3
3.2 Résolution . . . 3
3.2.1 Modélisation . . . 3
3.2.2 Génération . . . 3
3.2.3 Représentations des contraintes . . . 5
3.3 Résultats . . . 5

4 Jeu Singles **7**
4.1 Règles du jeu . . . 7
4.2 Résolution . . . 7
4.2.1 Modélisation . . . 7
4.2.2 Génération . . . 7
4.2.3 Représentations des contraintes . . . 7
4.3 Résultats . . . 8

5 Conclusion **8**

1 Introduction

Ce rapport présente la modélisation de deux jeux de logiques, "Unruly" et "Singles", de manière automatique à l'aide d'un solveur (basé sur des méthodes d'optimisation en programmation linéaire).

2 Solveur utilisé

La résolution se fait grâce au solveur "CPLEX". "IBM ILOG CPLEX Optimization Studio (often informally referred to simply as CPLEX) is an optimization software package." (source : Wikipedia)

3 Jeu Unruly

3.1 Règles du jeu

Le jeu Unruly consiste a combler une grille de taille n*m (avec n et m deux entiers pairs supérieurs ou égaux à 6) avec des tuiles soit noires soit blanches.
Le joueur est confronté à une grille pré-remplie (densité de remplissage initiale variable) et doit la combler en vérifiant les règles suivantes :

1. Il ne doit y avoir aucun alignement de trois tuiles successives de même couleur horizontalement
2. Il ne doit y avoir aucun alignement de trois tuiles successives de même couleur verticalement
3. Chaque ligne doit contenir autant de tuiles blanches que noires
4. Chaque colonne doit contenir autant de tuiles blanches que noires

Vous trouverez en figure 1 une instance du jeu ainsi que sa résolution.

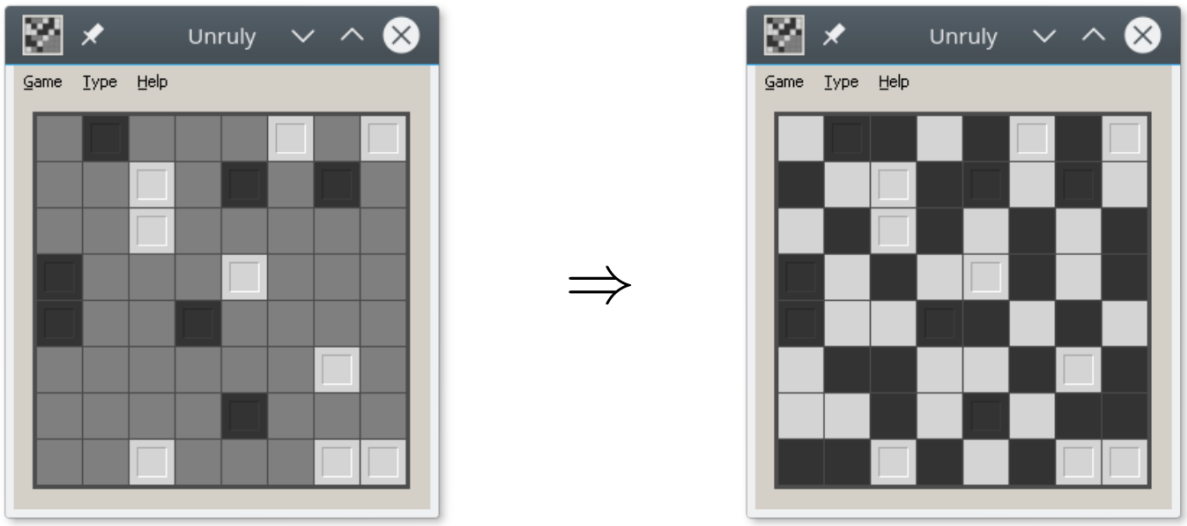


FIGURE 1 – Instance du jeu Unruly pour une grille de taille 8x8 et proposition d'une solution.

3.2 Résolution

3.2.1 Modélisation

Soit une grille du jeu de taille n x m. Nous avons adopté deux représentations :

1. Pour l'affichage et la génération d'instance : une matrice de taille n x m , remplie de 1 ou de 2 (1 signifiant une tuile blanche, 2 une tuile noire)
2. Pour le solveur en programmation linéaire : la grille est représentée par un vecteur binaire de dimension 3 que l'on notera $a(i, j, k)$ avec $i \in [1, n]$ et , $j \in [1, m]$ et $k \in [1, 2]$. Les coordonnées i et j représentent respectivement la ligne et la colonne de la case considérée et la coordonnée k représente le couleur de cette case. Avec k=1 pour une tuile blanche, et k=2 pour une tuile noire.

Il y a donc eu une partie "relecture" dans la partie "resolution.jl" et "io.jl" , pour passer d'une représentation à une autre.

3.2.2 Génération

Des commentaires de génération sont disponbiles dans le fichier texte generation-comments.txt".
On génère des grilles en choisissant en paramètre d'entrée la taille de la grille et la densité de remplissage.
Les instances sont générées de la manière suivante :

1. vérifier que la grille est bien paire x paire
2. vérifier que les dimensions sont supérieures à 6

3. on démarre une boucle de laquelle on sort : soit si on a rempli la grille avec la densité souhaitée, soit si on à atteint un nombre trop elevée d'itérations sans réussir à remplir la grille
4. sélecton d'une case aléatoire
5. choix d'une couleur aléatoire
6. on vérifie qu'on est pas en train de remplir la dernière case d'une ligne ou colonne via la fonction "$isalmostfull$
7. test si on peut remplir la case choisie avec la couleur choisie sans aligner trois cases (ce test se fait via la fonction "$threewillalign$"
8. le cas échéant on remplit la case

Les instances Voici la définition de la fonction :

```
Generate one POTENTIALLY solvable instance of the problem.
"potentially solvable" as in : the instance generated breaks no rules.
NB : it does not necessaraly mean that the instance can be solved

Arguments
- n: number of lines of the grid
- m : number of columns of the grid
- density: percentage in [0, 1] of initial values in the grid

NB : I expect the proportion of TRULY solvable instance genereated by this method
to be high when the density is low (under 0.4) but low when the density is high.
That is because we only make sure no rules are currently broken when we generate
the instance, not that it is truly solvable.
```

3.2.3 Représentations des contraintes

On répresente les contraintes sous les formes suivantes :

```
### Contraintes

#chaque case initialisée doit être conservée
for i in 1:l
    for j in 1:c
        for k in 1:2
            if t[i,j]==k
                @constraint(m,x[i, j, k] == 1)
            end
        end
    end
end

#chaque case a une valeur unique
for i in 1:l
    for j in 1:c
        @constraint(m,sum(x[i, j, k] for k in 1:2) == 1)
    end
end

# pas plus de 2 cases identiques ajdacentes horizontalement
for ligne in 1:l
    for colonne in 2:(c-1)
        for k in 1:2
            @constraint(m, sum(x[ligne, j, k] for j in (colonne-1):(colonne+1)) <= 2)
        end
    end
end

#pas plus de 2 cases identiques adjacentes verticalement
for colonne in 1:c
    for ligne in 2:(l-1)
        for k in 1:2
            @constraint(m, sum(x[i, colonne, k] for i in (ligne-1):(ligne+1)) <= 2)
        end
    end
end

#autant de cases blanches que de cases noires au sein d'une ligne
for ligne in 1:l
    @constraint(m, sum(x[ligne,j,1] for j in 1:c) == sum(x[ligne,j,2] for j in 1:c))
end

#autant de cases blanches que de cases noires au sein d'une colonne
for colonne in 1:c
    @constraint(m, sum(x[i,colonne, 1] for i in 1:l) == sum(x[i,colonne, 2] for i in 1:l))
end
```

FIGURE 2 – Contraintes pour le jeu unruly

3.3 Résultats

Voir "perfunruly.pdf" (diagramme de performance) ainsi que "unrulysolve.pdf" (temps de résolution et optimalité de la grille obtenue).
Le diagramme de performance laisse supposer que les instances de petites tailles sont résolues de manière quasi instantanées.

L'analyse du temps de résolution en fonction du de la taille (basé sur le doc unrulysizetime.pdf)donne l'allure suivante :

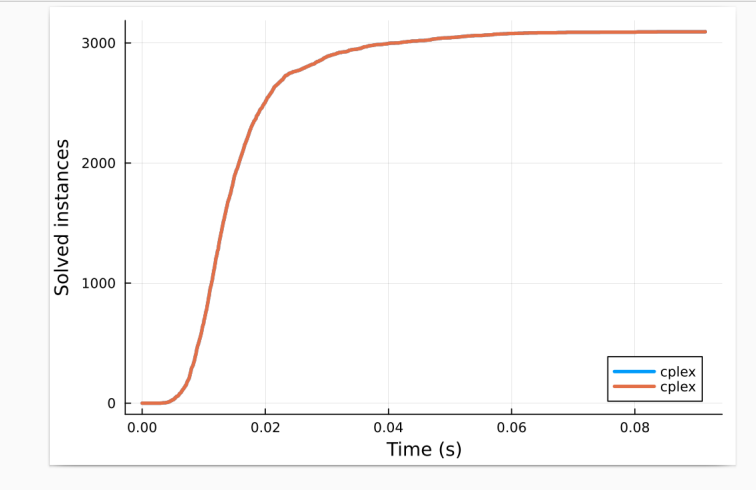
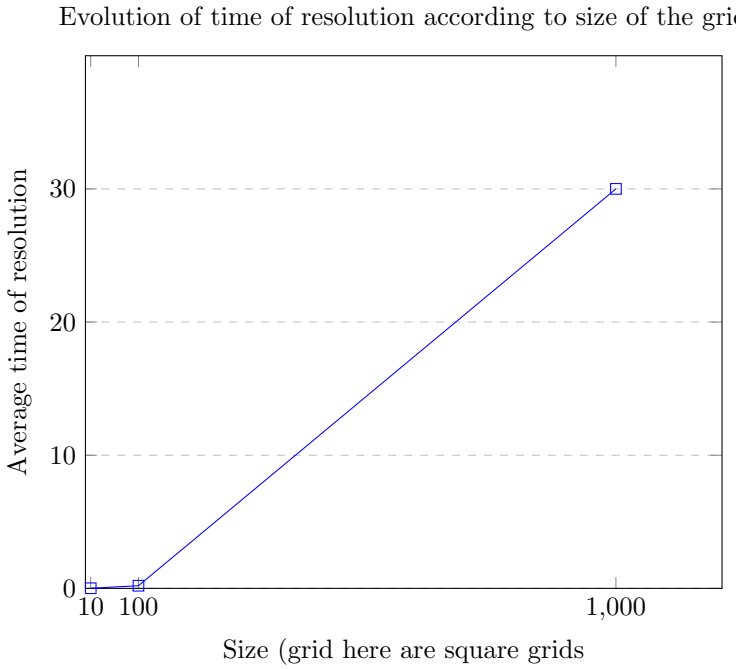


FIGURE 3 – Diagramme de performance pour le jeu unruly



NB : Impossible de generer une instance de taille 10 000, mon code n'est peut être pas assez optimisé. –¿ A discuter lors de la soutenance.