

Projet Phineloops : Rapport d'expérimentation

Betty Bismuth et Jean Clemenceau

Université Paris-Dauphine

M1 MIAGe

Java Avancé

Janvier 2018

Sommaire

1	Choix de la pièce à tester pour le solveur	3
2	Générateur avec choix du nombre de composantes connexes	3

1 Choix de la pièce à tester pour le solveur

Nous avons codé deux choix différents. Pour rappel, le premier `solveRandom` choisit au hasard la pièce par laquelle commencer la résolution puis avance dans la grille de gauche à droite. Le second `solveFix` récupère toutes les pièces fixées de la grille. Ces pièces sont les pièces vides et les croix c'est-à-dire celles qu'il n'est pas nécessaire de tester. Il va ensuite prendre une de ces pièces, récupérer ses voisins et trouver leurs orientations optimales. Puis la méthode effectue le même traitement avec tous les voisins de toutes les pièces fixées et termine les pièces restantes de gauche à droite.

Le choix à partir des pièces fixées de la grille présente une nette amélioration algorithmique. En effet, on peut constater que le temps d'exécution sur de grandes grilles 32x32 par exemple est court (environ 8 secondes) alors qu'avec un choix aléatoire puis en allant de gauche à droite, on obtient un temps significativement plus long. Cela s'explique par le fait qu'en débutant par les cases fixées, on fait évoluer la solution autour de pièces dont la position est certaine. Cela diminue donc le nombre de retours en arrière, et donc la complexité de l'algorithme.

2 Générateur avec choix du nombre de composantes connexes

Nous avons codé un générateur de grille dans laquelle le nombre de composantes connexes est déterminé par l'utilisateur. Nous avons choisi d'utiliser une récursion afin de colorer toute la grille et de faire apparaître les différentes composantes connexes. De plus, il a fallu limiter le choix de l'utilisateur. Nous avons d'abord pensé que le nombre de composantes connexes maximal serait le nombre de pièces total de la grille divisé par 2. En effet, il faut au moins deux pièces pour constituer une composante connexe. Cependant, plus le nombre de composantes connexes demandé est élevé, plus l'exécution est ralentie. Nous avons donc décidé de limiter le nombre de composantes connexes au tiers du nombre de pièces total de la grille.