

INFO0054-1 : Programmation fonctionnelle

Projet : Puzzles réguliers

Maxime MEURISSE (s161278)

François ROZET (s161024)

19 avril 2019

1 Solver

Notre solver se base sur l'algorithme BFS. Ce dernier ne permet pas d'obtenir toutes les solutions d'un problème régulier, seulement les plus courtes, c.-à-d. celles qui ont été les « premières » à visiter leurs états.

Dans la méthode originale, à chaque étape, les chemins dont le dernier état a déjà été visité précédemment sont supprimés. Dans notre implémentation, nous avons décidé de les garder en mémoire. Lorsque une solution est trouvée par l'algorithme, nous réintroduisons tous les chemins dont l'état final est un état intermédiaire de la solution.

De cette façon nous sommes assurés de trouver toutes les solutions et dans l'ordre désiré, c.-à-d. de la plus courte à la plus longue.

Néanmoins, cette méthode est loin d'être optimale. Notamment, elle ne réutilise pas la solution trouvée pour en former de nouvelles à partir des chemins mis de côté. En effet, puisque certains de ces derniers ont un état commun avec la solution, il devrait être possible d'en construire une autre au minimum aussi longue.

3 Heuristique

Le solver « avec heuristique » implémente une *priority queue* dont la priorité est la somme de l'heuristique au dernier état du chemin et de la longueur de ce dernier. L'addition de la longueur permet de privilégier les chemins courts.

Nous avons implémenté les deux heuristiques de taquin H_1 et H_2 . En mesurant leur temps d'exécution pour trouver un grand (100) nombre de solutions, il est apparu clairement que la seconde était plus rapide (15s contre 2000s) . C'est donc cette dernière que nous avons désignée comme heuristique.