

Laboratoire #2
Résolution d'un puzzle
Version 1.0

Patrick Cardinal

Durée : 3 semaines

Introduction

Les fonctions récursives du type « retour en arrière » ou « *backtracking* » peuvent être utilisées pour résoudre différents types de problèmes d'optimisation combinatoire. Dans le cadre de ce laboratoire, la technique sera utilisée pour résoudre un puzzle. Le puzzle se joue normalement sur un plateau contenant un certain nombre de trous. Au départ, un certain nombre de trous sont bouchés par des tiges, mais d'autres sont libres. La figure 1 suivante représente une configuration possible, le cercle rouge indiquant la position libre. Le but du puzzle est d'éliminer toutes les tiges sauf une.

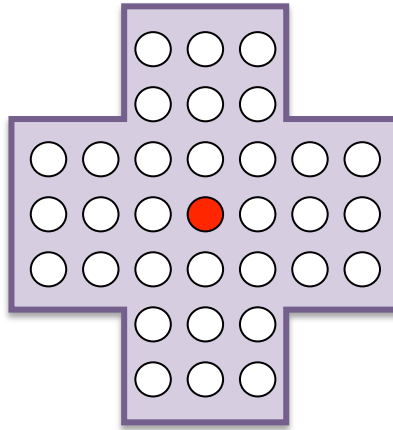


Figure 1 – Plateau d'un puzzle

Les règles sont simples :

- Le seul mouvement possible est le saut par-dessus une tige si la destination est libre;
- Lors d'un saut par dessus une tige, la tige en question est éliminée du plateau;
- Il y a 4 directions possibles : haut, bas, gauche, droite

Au départ, il n'y a aucune information disponible permettant de faire un choix du meilleur saut. Comme chaque mouvement mène à d'autres possibilités de mouvements, le nombre de possibilités augmente rapidement. Noter que plusieurs solutions peuvent être possibles. Il est aussi possible qu'il n'y ait aucune solution pour un puzzle donné.

Le but du laboratoire est de concevoir un algorithme de type « retour en arrière » qui sera en mesure de résoudre ce type de puzzle.

Les objectifs de ce laboratoire sont :

- Se familiariser avec les algorithmes de type « retour en arrière »
- Implémentation efficace de l'algorithme
- Optimisation

Programme

Pour résoudre ce problème, vous devez concevoir un algorithme de type retour en arrière (« *backtracking* »). Cet algorithme consiste à effectuer des déplacements de tiges jusqu'à ce qu'il n'y ait plus de mouvement possible. S'il reste seulement une tige de disponible, le puzzle est résolu. Dans le cas contraire, il faut retourner en arrière afin d'essayer une autre alternative. L'algorithme continue de cette façon jusqu'à ce qu'une solution soit trouvée ou que toutes les combinaisons possibles aient été essayées.

Il y a 3 fonctions importantes à implémenter. La première fonction est la lecture du fichier d'entrée dans lequel le puzzle à résoudre est décrit. Le format de ce fichier est le suivant. La dimension maximale des plateaux est de 7x7. Il y aura donc un maximum de 49 trous. Le plateau est décrit par une suite de chiffres. Les cases inaccessibles (en dehors de la surface de jeu) sont représentées par le chiffre 0. Les trous contenant une tige sont représentés par le chiffre 1 et les trous libres par le chiffre 2. Par exemple, le fichier correspondant au puzzle de la figure 1 serait :

```
0011100
0011100
1111111
1112111
1111111
0011100
0011100
```

La deuxième fonction sert à déterminer les coups valides à partir d'une position donnée. Cette fonction vous permettra de déterminer les mouvements à essayer lors de la recherche d'une solution.

La troisième fonction implémente l'algorithme de retour en arrière qui permet de résoudre le puzzle spécifié par l'utilisateur. Cette fonction devrait retourner vrai si une solution a été trouvée et faux si aucune solution n'est possible.

Autres spécifications

La sortie de votre programme devra contenir les trois éléments suivants :

1. la liste des mouvements pour obtenir la solution,
2. la solution au puzzle spécifié,
3. le nombre de nœuds explorés lors de la recherche,
4. le temps d'exécution.

Votre programme devra aussi avertir l'utilisateur si le puzzle n'a pas de solutions.

Vérification du programme

À la date déterminée par le responsable du laboratoire, vous devrez faire la démonstration que votre programme fonctionne. À ce moment, le responsable du laboratoire vous fournira différents fichiers de test afin de vérifier le bon fonctionnement de votre programme et d'évaluer son efficacité. Les puzzles utilisés pour tester votre programme comprendront entre 15 et 31 tiges. Il peut y avoir des puzzles impossibles à résoudre.

Un programme de démonstration ainsi que des fichiers de test seront disponibles sur le site Moodle du cours.

Rapport de laboratoire

Votre rapport de laboratoire devra contenir les éléments suivants :

- Description de votre algorithme
- Améliorations apportées à l'algorithme de base pour améliorer la performance
- Autres commentaires pertinents, s'il y a lieu

Votre rapport doit avoir un maximum de 4 pages. Toutes les pages supplémentaires seront ignorées lors de la correction.

Barème de correction

Le barème de correction est le suivant :

Fonctionnement de base	3
Fonctionnement avec le(s) test(s)	2
Performances (nombre de nœuds explorés)	2
Rapport de laboratoire	3
Total	10

Le barème concernant les points de performance sera expliqué par le responsable du laboratoire.