

UNIVERSITÉ LIBRE DE MONTRÉAL

Rapport : TP1 - Sudoku

Pierre Gérard, Alexandre Savard

IFT 3335 Intelligence artificielle : Introduction

Jian-Yun Nie, William Lechelle

Année académique 2015 - 2016

Table des matières

1	Formulation du problème	2
2	Search	2
3	Hill-climbing	2
3.1	Implémentation	2
3.2	Résultat	2
3.3	Analyse des résultats	3
4	Heuristic search	4
4.1	Implémentation	4
4.2	Résultat	4
4.3	Analyse des résultats	4
5	Comparaison	4

1 Formulation du problème

2 Search

3 Hill-climbing

3.1 Implémentation

L'algorithme a été implémentée de manière à ce que la grille initiale soit remplie de manière totalement aléatoire. L'état initiale est donc cette grille initiale. On peut le visualiser comme la racine d'un arbre. Il a 9 enfants. Chaque enfant correspond à la permutation de (case1,case2) dans un des carré de dimension 3x3 telle que cette permutation minimise le nombre de conflit global. la permutation optimale est trouvé en les testant toutes à l'intérieur d'un carré 3x3

Cela permet d'affirmer que la probabilité que l'algorithme trouve une solution optimale tend vers 1 pour un très grand nombre d'essai avec des configurations initiales différentes.

3.2 Résultat

Ci dessous un exemple de l'exécution du Hill-Climbing.

```
--- INPUT ---
. . 3 | . 2 . | 6 . . |
9 . . | 3 . 5 | . . 1 |
. . 1 | 8 . 6 | 4 . . |
-----
. . 8 | 1 . 2 | 9 . . |
7 . . | . . . | . . 8 |
. . 6 | . . 8 | 2 . . |
-----
. . 2 | . . 9 | 5 . . |
8 . . | 2 . 3 | . . 9 |
. . 5 | . 1 . | 3 . . |
-----
```

```
Hill climbing
4 8 3 | 9 2 1 | 6 7 5 |
9 6 7 | 3 4 5 | 8 2 1 |
2 5 1 | 8 7 6 | 4 9 3 |
-----
5 4 8 | 1 6 2 | 9 3 7 |
7 2 9 | 5 3 4 | 1 6 8 |
3 1 6 | 7 9 8 | 2 5 4 |
-----
1 3 2 | 6 8 9 | 5 4 6 |
8 7 4 | 2 5 3 | 7 1 9 |
6 9 5 | 4 1 7 | 3 8 2 |
-----
```

ECHEC, malgre 20 test de configuration initiale differente
Ci dessus, un maximum local (4 conflits restant)

Comme indiqué, il reste 4 conflits et la solution n'est pas optimal. Pour des grilles plus simple l'algorithme trouve la solution comme on peut voir ci-dessous.

```
--- INPUT ---
```

```

. . . | 9 2 . | . 5 7 |
9 . 7 | . 4 5 | 8 2 1 |
2 5 1 | 8 7 6 | 4 9 3 |
-----
5 4 8 | 1 . 2 | 9 7 6 |
7 2 9 | 5 6 4 | 1 3 8 |
1 3 . | 7 9 8 | 2 4 5 |
-----
3 7 2 | 6 8 9 | 5 1 4 |
8 1 4 | 2 5 3 | 7 6 9 |
. 9 5 | 4 1 7 | 3 8 2 |
-----

```

Hill climbing

```

4 8 3 | 9 2 1 | 6 5 7 |
9 6 7 | 3 4 5 | 8 2 1 |
2 5 1 | 8 7 6 | 4 9 3 |
-----
5 4 8 | 1 3 2 | 9 7 6 |
7 2 9 | 5 6 4 | 1 3 8 |
1 3 6 | 7 9 8 | 2 4 5 |
-----
3 7 2 | 6 8 9 | 5 1 4 |
8 1 4 | 2 5 3 | 7 6 9 |
6 9 5 | 4 1 7 | 3 8 2 |
-----

```

Succes !

Pour cette grille simple, Hill climbing a bien trouvé la solution optimale.

3.3 Analyse des résultats

Il n'est pas surprenant que Hill Climbing, un algorithme de recherche local, ne trouve pas de solution optimale. En effet, il est basé sur des changements locaux et trouve régulièrement de bonne solution mais rarement la solution optimale. La figure suivante illustre cela.

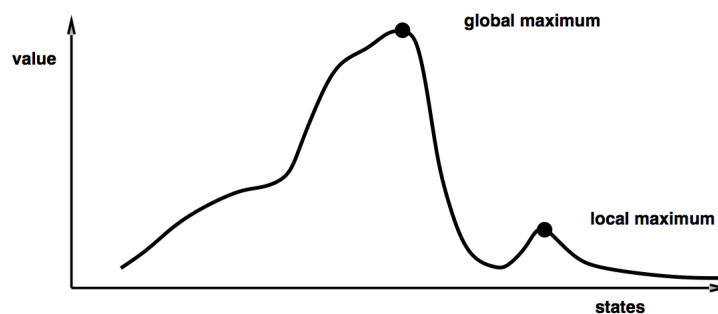


FIGURE 1 – Representation du maximum local

Une fois au maximum local, notre algorithme s'arrête comme il n'a pas de meilleur voisin. Il est incapable de dire si il existe un meilleur sommet.

Une meilleure technique serait le Simulated annealing, non demandé ici.

4 Heuristic search

4.1 Implémentation

4.2 Résultat

4.3 Analyse des résultats

5 Comparaison