

Rapport du TP1 de IFT 3335

Par : Mike useni et Soumaila Keita

Introduction

Les tests de performance de ces TP sont effectués sur les ensembles de cas des Sudoku compris dans les fichiers 100sudoku.txt, top95.txt et 1000sudoku.txt, ainsi qu'un ensemble de 100 configurations de sudoku généré aléatoirement.

Les programmes execute sont sudoku_Q1.py, sudoku_Q2.py, sudoku_Q3.py, sudoku_Q4.py, sudoku_Q5.py.

Question 2 : comparaison des performances avec et sans le 3^e critère de Norvig

(1*) Rendu de l'exécution du programme (sudoku_Q1) avec le 3^e critères de Norvig.

```
"C:\Users\mike useni\Anaconda3\envs\TP1 IA\python.exe" "C:/Users/mike useni/Desktop/IA
All tests pass.
Solved 100 of 100 100sudoku puzzles (avg 0.00 secs (332 Hz), max 0.01 secs).
Solved 95 of 95 top95 puzzles (avg 0.01 secs (102 Hz), max 0.05 secs).
Solved 1000 of 1000 1000sudoku puzzles (avg 0.00 secs (351 Hz), max 0.01 secs).
Solved 99 of 99 random puzzles (avg 0.00 secs (352 Hz), max 0.00 secs).

Process finished with exit code 0
```

(2*) Rendu de l'exécution du programme (sudoku_Q2) sans le 3^e critères de Norvig.

```
"C:\Users\mike useni\Anaconda3\envs\TP1 IA\python.exe" "C:/Users/mike useni/Desktop/I
All tests pass.
Solved 100 of 100 100sudoku puzzles (avg 0.00 secs (266 Hz), max 0.01 secs).
Solved 95 of 95 top95 puzzles (avg 0.01 secs (80 Hz), max 0.14 secs).
Solved 1000 of 1000 1000sudoku puzzles (avg 0.00 secs (311 Hz), max 0.03 secs).
Solved 99 of 99 random puzzles (avg 0.00 secs (362 Hz), max 0.00 secs).

Process finished with exit code 0
```

Tableau récapitulatif

	(1*) sudoku_Q1			(2*) sudoku_Q2		
	avg	Temps max	Taux de réussite	avg	Temps max	Taux de réussite
100sudoku	0.00	0.01	100%	0.00	0.01	100%
Top95	0.01	0.05	100%	0.01	0.14	100%
1000sudoku	0.00	0.01	100%	0.00	0.03	100%
random	0.00	0.00	100%	0.00	0.00	100%

On peut observer que même si (1*) sudoku_Q1 et (2*) sudoku_Q2 ont le même **avg** et tout deux un **taux de réussite** de 100%, (1*) sudoku_Q1 a un **temps max** plus faible en générale. (1*) sudoku_Q1 est donc le plus performant parce qu'il est plus rapide.

Question 3: choix d heuristique (sudoku Q3)

Pour cette partie nous avons étudié 2 heuristiques, **Candidats uniques cachés** et **Paires nues** . Nous avons finalement opté pour le **Paires nues** parce le cas d'utilisation du **Candidats uniques cachés** n arrive que très rarement ce qui réduit le risque d'optimiser les algorithmes.

L'heuristique **Paires nues** recherche 2 cellule qui ont une paire de possibilité identiques sur la même ligne, les fixe en les retirant comme possibilité du reste de la ligne ce qui réduit les possibilités.

Question 4: l'algorithme de Hill-climbing avec swap (sudoku Q4)

L'algorithme de Hill-climbing a un taux de réussite 0% et même couple avec un swap il ne résout pas tous les conflits rencontrés.

Une fois exécuté le programme retourne nombre de conflits initialement rencontrés (**Nb_conflits_In**) et le nombre de conflits restant après exécution du swap (**Nb_conflits_Ameliorer**).

Exemple d'exécution sur grid2

```
"C:\Users\mike useni\Anaconda3\envs\TP1 IA\python.exe" "C:\
Nb_conflits_In:
158
Nb_conflits_Ameliorer
28

Process finished with exit code 0
```

On trouve 158 conflits avant l'exécution de l'algorithme et 28 après les swaps.