

Etape 3 : Aide à la décision

Rapport Projet

CHEUCLE Pierre – HUYNH Pascal – ODIER Valentin – WAKIM Marie

24/05/2012

Contenu

Introduction	2
Rappel de notre structure.....	3
Les stratégies	3
Super_resolution_propagation_contraintes2	3
Super_resolution_propagation_contraintes5	4
Super_resolution_propagation_contraintes8	4
Super_resolution_propagation_contraintes4	4
Etude comparative des résultats	4
Notre analyse.....	4
Analyse des tests :	5

Introduction

Ce projet nous a amenés à réaliser un algorithme permettant de résoudre un sudoku. Pour parvenir à cela nous avons dû imaginer différentes stratégies de recherche telles que :

- *Vérification des possibilités case par case*
- *Affectée la case avec le moins de possibilités*
- *Affectée la case avec le plus de possibilités*
- *Affectée la case avec le moins d'influence sur la suite*
- *...*

Dans une première partie nous rappellerons la structure de données que nous avons choisie, puis nous expliquerons les stratégies développées.

Rappel de notre structure

Nous avons gardé la même structure de données que dans le TP1 :
Par exemple, pour cette grille de sudoku :

4	2		
			1
2			

En utilisant un vector de vector d'entiers. (`Vector<Vector<int>>`), nous avons :

↓	↓	↓	↓
0	4	0	2
0	2	0	0
0	0	0	0
0	0	1	0

Les stratégies

Dans cette partie nous allons développer les stratégies mises en place. Les noms des fonctions sont ceux du programme que nous simplifions par « fonctionX » (avec X le numéro correspondant).

Ci-dessous nous avons simplement retenu les fonctions dont la stratégie est la plus intéressante :

Super_resolution_propagation_contraintes2

La fonction2 a été expliquée dans le tp2 : pour chaque cases vides de la grille de Sudoku, nous calculons le nombre de valeurs possibles. Puis nous choisissons d'affecter la case avec le moins de possibilités.

Cette fonction n'est pas très optimisée, mais nous a été utile pour réaliser les stratégies suivantes.

Super_resolution_propagation_contraintes5

Dans la fonction5, nous affectons une case, puis nous analysons l'impact de ce changement sur les autres cases du sudoku.

Nous conservons alors la case qui aura le moins d'impact sur le reste du sudoku.

Super_resolution_propagation_contraintes8

Dans la fonction 8, nous utilisons le principe utilisé pour la fonction5, nous choisissons d'affecter la case qui :

- A le moins de possibilités
- Aura le plus d'impact sur le reste du sudoku.

Super_resolution_propagation_contraintes4

La fonction4 est semblable à la fonction8, nous affectons la case qui :

- a le moins de possibilités
- aura le moins d'impact sur le reste du sudoku.

Etude comparative des résultats

Notre analyse

Nous avons effectué des tests pour chaque fonction avec différentes grilles de sudoku 3x3. Ces tests nous ont permis de trouver les fonctions les plus et moins performantes.

Pour chaque fonction, nous relevons :

- L'état de changement : le nombre de nœuds parcourus dans l'arbre
- Le nombre de retour : le nombre de remontée dans l'arbre
- Le temps d'exécution : le temps d'exécution de l'algorithme

Nous avons testé chaque fonction sur 14 Sudokus différents :

- Sur 9 sudokus pré-remplis
- Les suivants sont des sudoku vides : 2x2, 3x3, 4x4, 5x5 et 6x6.

Nous obtenons les résultats suivants :

Tableau excel en annexe.

Analyse des tests :

Notre première observation permet de voir que les fonctions 6 et 10 sont inefficaces au vue des résultats aberrants obtenus. Nous ne les utiliserons donc pas pour comparer les résultats : les stratégies utilisées ne sont pas efficaces.

Donc analyse entre les fonctions 2, 4, 5, 7 et 8 :

Nous constatons que la fonction 2 est la fonction qui s'exécute la plus rapidement pour la plupart des sudokus (sudokus 1, 2, 3, 4, 8 et 11).

A l'inverse, la fonction 5 est la fonction qui s'exécute la plus lentement pour la plupart des sudokus (sudokus 1, 2, 3, 4, 7, 9, 11) .

Les fonctions 7 et 4 apparaissent aussi comme une bonne fonction.

En annexe nous avons fourni des graphes affichant les états, le nombre de retour, le temps d'exécution en fonction de nos fonctions. Nous visualisons très clairement que les fonctions 2 et 4 sont systématiquement meilleurs quel que soit le sudoku choisi.

Finalement, il est difficile de conclure sur quelle est la fonction la plus performante. Cependant des fonctions apparaissent plus performantes que d'autres : les 2, 4, 7 et 8. La fonction 4 semble avoir une puissance supérieure, sa moyenne étant plus basse que les autres. Nous pouvons supposer que pour des sudoku plus grand, elle sera plus efficace que les autres.

Par ailleurs, nous observons qu'une fonction peut parcourir un nombre d'états important et s'exécuter rapidement : ceci étant lié à l'algorithme.

Annexes

	Colonne1	Fonction 2	Fonction 4	Fonction 5	Fonction 8	Fonction 6	Fonction 7	Fonction 9	Fonction 10
Sudoku 1	Etat Chargement	47	51	121	57	256362	121	322	648
	Retour	0	3	19	4	60005	19	88	130
	Execution	62	63	94	78	157764	94	577	1154
	Total	109	117	234	139	474131	234	987	1932
Sudoku 2	Etat Chargement	252	1178	16638	6468	0	16638	77519	216783
	Retour	12	223	4126	1189	0	4126	25002	55983
	Execution	421	1794	12105	9313	0	12168	12291	378162
	Total	685	3195	32869	16970	X	32932	114812	650928
Sudoku 3	Etat Chargement	53	206	1577	282	0	1577	15569	56897
	Retour	0	26	316	28	0	316	5142	12923
	Execution	62	281	889	390	0	889	22199	96003
	Total	115	513	2782	700	X	2782	42910	165823
Sudoku 4	Etat Chargement	264	471	787	437	0	787	5038	0
	Retour	16	85	134	52	0	134	1282	0
	Execution	374	686	515	671		531	7285	
	Total	654	1242	1436	1160	X	1452	13605	X
Sudoku 5	Etat Chargement	4835	3102	4126	43746	0	4126	140615	0
	Retour	723	733	4084	8265	0	1084	0	0
	Execution	8097	6053	2792	62962	0	2823	0	0
	Total	13655	9888	11002	114973	X	8033	140615	X
Sudoku 6	Etat Chargement	166	97	94	108		94	708	
	Retour	8	11	13	15	0	13	277	0
	Execution	249	156	63	187	0	47	1435	0
	Total	423	264	170	310	X	154	2420	X

Etape 3 : Aide à la décision

CHEUCLE Pierre – HUYNH Pascal– ODIER Valentin – WAKIM Marie

Sudoku 7	Etat Chargement	55	54	148	54	0	148	98	0
	Retour	1	1	22	1	0	22	12	0
	Execution	78	78	93	78	0	93	109	0
	Total	134	133	263	133	X	263	219	X
Sudoku 8	Etat Chargement	46	55	78	66	0	78	98	0
	Retour	0	6	8	7	0	8	12	0
	Execution	47	62	47	78	0	32	109	0
	Total	93	123	133	151	X	118	219	X
Sudoku 9	Etat Chargement	217	140	10296	234	0	10296	38884	0
	Retour	8	15	2376	26	0	2376	11583	0
	Execution	296	172	5819	234	0	5834	48158	0
	Total	521	327	18491	494	X	18506	98625	X
Sudoku 10	Etat Chargement	16	16	16	16	0	16	16	0
	Retour	0	0	0	0	0	0	0	0
	Execution	16	0	15	0	0	0	0	0
	Total	32	16	31	16	X	16	16	X
Sudoku 11	Etat Chargement	81	98	391	111	0	274	364	0
	Retour	0	8	136	8	0	89	99	0
	Execution	172	203	437	187	0	218	795	0
	Total	253	309	964	306	X	581	1258	X
Sudoku 12	Etat Chargement	282	256	256	0	0	0	0	0
	Retour	9	0	0	0	0	0	0	0
	Execution	2995	2948	936	0	0	0	0	0
	Total	3286	3204	1192	0	X	X	X	X
Sudoku 12	Etat Chargement	0	625	0	0	0	0	0	0
	Retour	0	0	0	0				

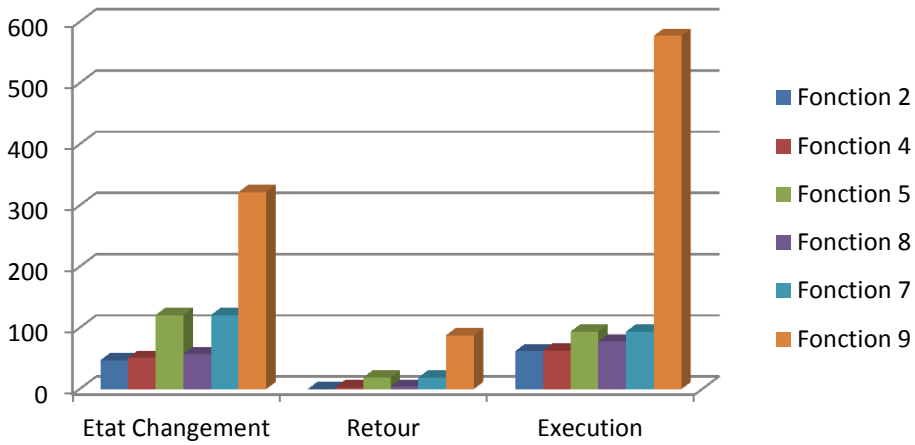
Etape 3 : Aide à la décision

CHEUCLE Pierre – HUYNH Pascal– ODIER Valentin – WAKIM Marie

	Execution	0	31044	0	0	0	0	0	0
	Total	X	31669	X	X	X	X	X	X
Sudoku 13	Etat Chargement	0	1296	0	0	0	0	0	0
	Retour	0	0	0	0	0	0	0	0
	Execution	0	215640	0	0	0	0	0	0
	Total	X	216936	X	X	X	X	X	X
Sudoku 14	Etat Chargement				0	0	0	0	0
	Retour					0	0	0	0
	Execution				0	0	0	0	0
	Total	X	X	X	X	X	X	X	X
TOTAL (ms)		19960	267936	69567	135352	474131	65071	415686	818683
TOTAL (s)		19,96	267,94	69,57	135,35	474,13	65,07	415,69	818,68

Sudoku 1

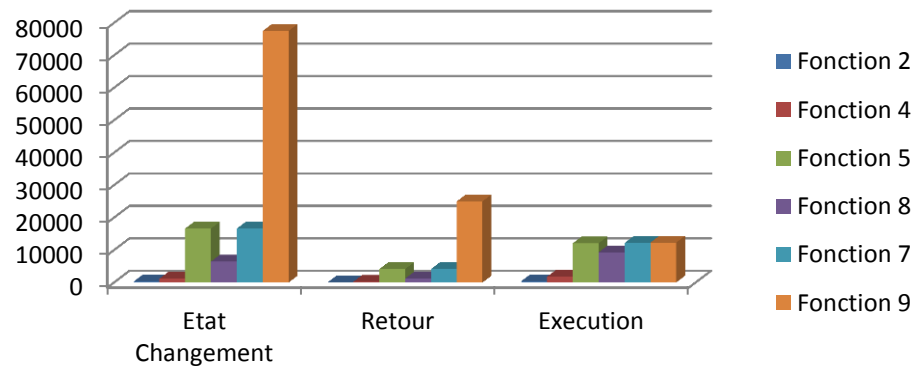
Etat	Fonction 2	Fonction 4	Fonction 5	Fonction 7	Fonction 9
Etat Changement	60	65	135	75	340
Retour	15	20	35	15	105
Execution	75	80	110	110	590



Sudoku 2

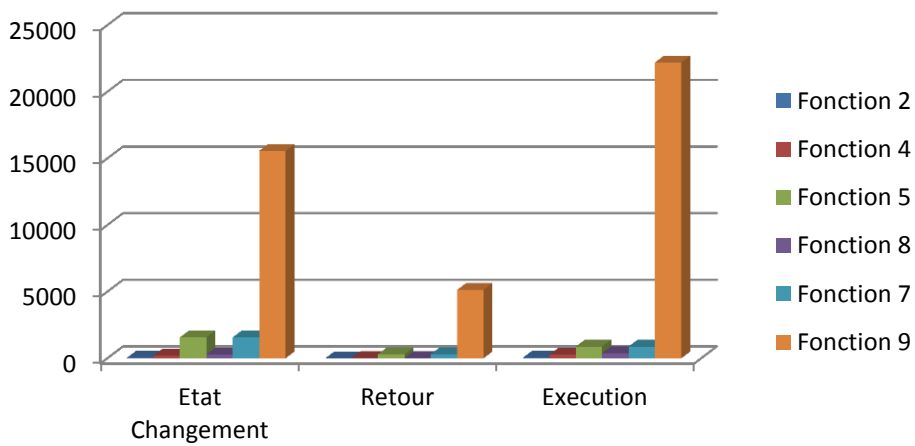
A bar chart titled "Sudoku 2" comparing the number of calls for six different functions across three states: "Etat Changement", "Retour", and "Execution". The Y-axis represents the number of calls, ranging from 0 to 80,000 in increments of 10,000. The legend identifies the functions by color: Foncton 2 (blue), Foncton 4 (red), Foncton 5 (green), Foncton 8 (purple), Foncton 7 (teal), and Foncton 9 (orange). Foncton 9 shows the highest number of calls in the "Etat Changement" state, while Foncton 5 and Foncton 7 show higher call counts in the "Retour" and "Execution" states.

Etat	Fonction 2	Fonction 4	Fonction 5	Fonction 8	Fonction 7	Fonction 9
Etat Changement	~2,000	~2,000	~18,000	~10,000	~18,000	~80,000
Retour	~2,000	~2,000	~6,000	~3,000	~6,000	~28,000
Execution	~2,000	~2,000	~14,000	~12,000	~14,000	~14,000

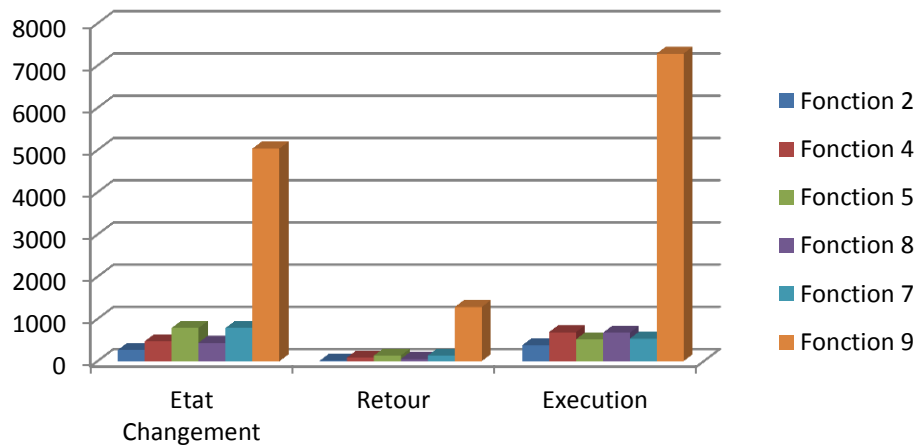


Sudoku 3

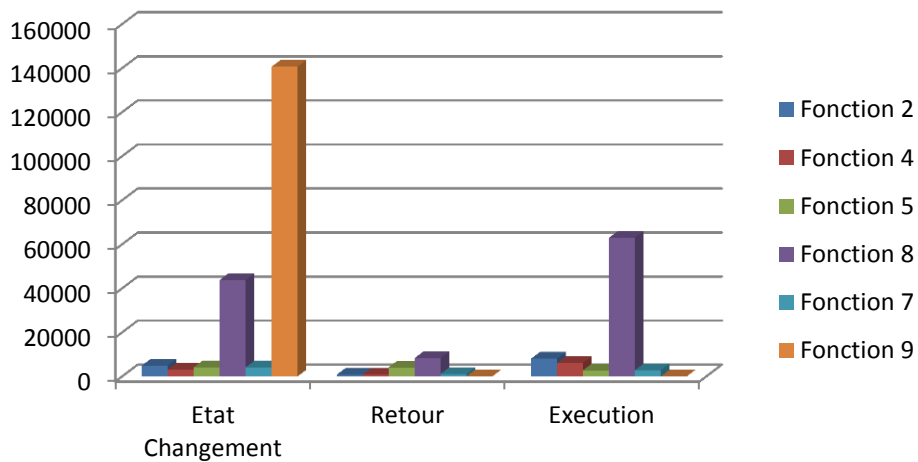
Etat	Fonction 2	Fonction 4	Fonction 5	Fonction 8	Fonction 7	Fonction 9
Etat Changement	~1000	~1000	~2000	~1000	~2000	~16000
Retour	~1000	~1000	~1000	~1000	~1000	~5500
Execution	~1000	~1000	~1500	~1000	~1500	~23000



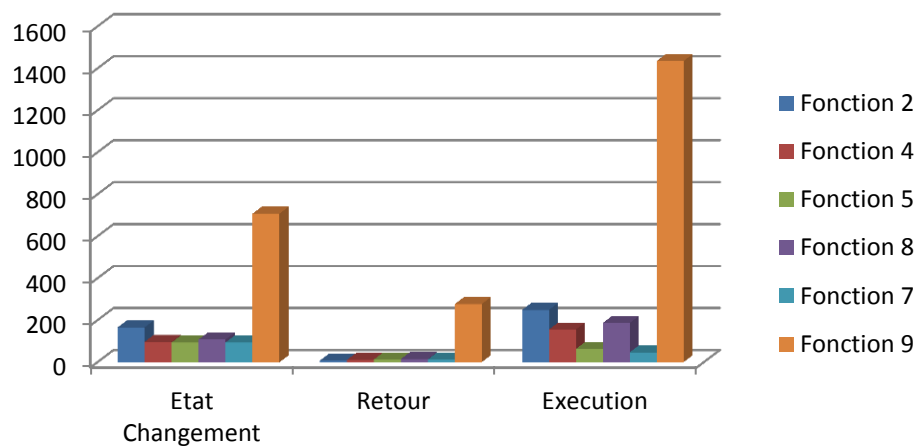
Sudoku 4



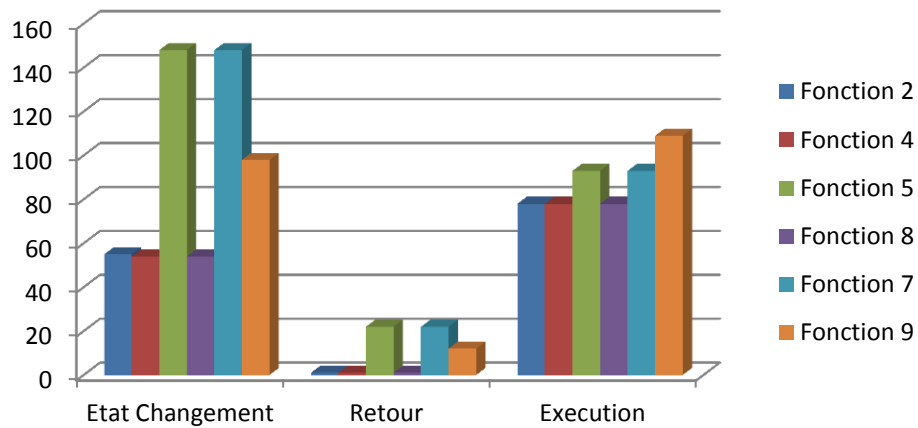
Sudoku 5



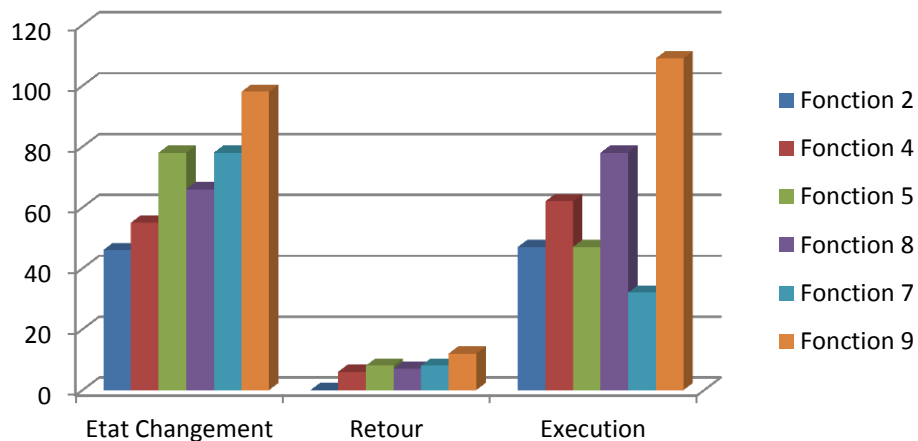
Sudoku 6



Sudoku 7



Sudoku 8



Sudoku 9

