



Présentation 7

PSAR

David TOTY
Maxime TRAN





Objectif:
Modéliser et résoudre le problème du Sudoku



Qu'est ce qu'un Sudoku ?

Un sudoku est un jeu de puzzle sous la forme d'une grille de taille $N^2 \times N^2$ composé de N^2 de lignes, N^2 de colonnes et N^2 de régions de taille $N \times N$.

Exemple:

- Sudoku classique:
 - ☐ $N=3$ correspond à la taille des régions
 - ☐ 9 lignes, 9 colonnes et 9 régions
 - ☐ 81 cases

Différents types de Sudoku

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

Sudoku classique

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

Chaos Sudoku

3		15			22	4	16	15
25		17						
		9			8	20		
6	14			17			17	
	13		20					12
27		6			20	6		
				10			14	
	8	16			15			
				13			17	

Killer Sudoku

Samurai Sudoku

[illegible]

Rappel: Programmation par contraintes

En programmation par contraintes, un problème doit être formulé à l'aide des notions suivantes :

- Des **variables**, des **domaines** et des **contraintes**.

Exemple:

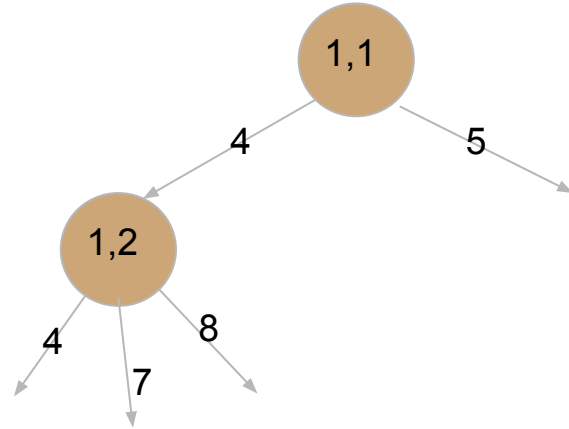
Variables: Les 81 cases du Sudoku

Domaines: L'ensemble {1, 2, 3, 4, 5, 6, 7, 8, 9}

Contraintes: 1 seule occurrence de valeur sur chaque ligne, colonne et région

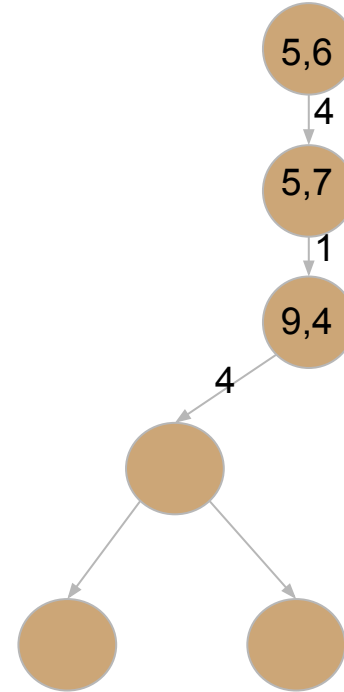
Algorithme naïf

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



Algorithme adaptatif

		3		2		6		
9			3		5			1
		1	8		6	4		
		8	1		2	9		
7					x	x		8
		6	7		8	2		
		2	6		9	5		
8			2		3			9
		5	x	1		3		



Les différents algorithmes de recherche

Codé sous Java:

- SolveNaif (int row, int col);
- SolveAdaptatif (int row, int col, int pos);
- SolveDynamique ();

Algorithme adaptatif

Tableau du Sudoku:

```
0 6 0 0 5 0 0 2 0
0 0 0 3 0 0 0 9 0
7 0 0 6 0 0 0 1 0
0 0 6 0 3 0 4 0 0
0 0 4 0 7 0 1 0 0
0 0 5 0 9 0 8 0 0
0 4 0 0 0 1 0 0 6
0 3 0 0 0 8 0 0 0
0 2 0 0 4 0 0 5 0
```

Tableau de parcours:

(nombre de degré de liberté, ligne, colonne)

1 : 3 : 7

1 : 6 : 4

1 : 8 : 4

2 : 0 : 4

2 : 0 : 6

2 : 2 : 4

2 : 2 : 6

2 : 3 : 5

2 : 4 : 1

2 : 4 : 7

...

Algorithme adaptatif

```
public void initParcours();
```

```
public void initFait();
```

Tableaux:

- Possibilité
- Etages

Algorithme adaptatif

```
public void initParcours();
```

```
public void initFait();
```

Tableaux:

- Possibilité
- Etages

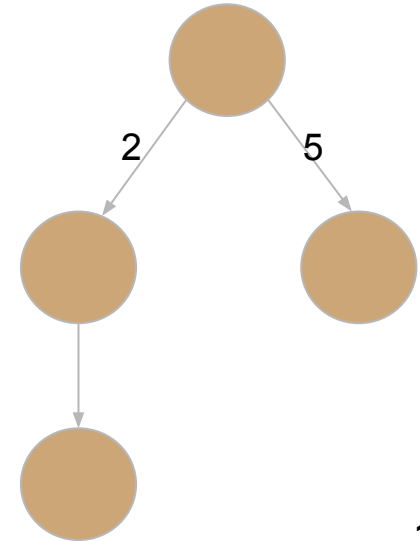
Tableau d'étages
stocke n.

n

Tableau de
possibilités
stocke la valeur
5.

n+1

n+2



Résultats & tests.

*** Résolutions de 300 Sudokus

Taille : N = 2 Indices = 5

Temps total algo 1 : 1609409 nanosecondes
Temps moyen algo 1 : 5364 nanosecondes
Temps max algo 1 : 59897 nanosecondes
Temps min algo 1 : 1077 nanosecondes
Variance algo 1 : 16933225 nanosecondes
Ecart type algo 1 : 4115 nanosecondes
Quartile 1 algo 1 : 2184 nanosecondes
Quartile 2 algo 1 : 2891 nanosecondes
Quartile 3 algo 1 : 4810 nanosecondes

* ***** *

Temps total algo 2 : 4289730 nanosecondes
Temps moyen algo 2 : 14299 nanosecondes
Temps max algo 2 : 121589 nanosecondes
Temps min algo 2 : 5042 nanosecondes
Variance algo 2 : 62188996 nanosecondes
Ecart type algo 2 : 7886 nanosecondes
Quartile 1 algo 2 : 7226 nanosecondes
Quartile 2 algo 2 : 10724 nanosecondes
Quartile 3 algo 2 : 14595 nanosecondes

*** Résolutions de 300 Sudokus

Taille : N = 2 Indices = 7

Temps total algo 1 : 1259320 nanosecondes
Temps moyen algo 1 : 4197 nanosecondes
Temps max algo 1 : 42200 nanosecondes
Temps min algo 1 : 1380 nanosecondes
Variance algo 1 : 3674889 nanosecondes
Ecart type algo 1 : 1917 nanosecondes
Quartile 1 algo 1 : 2941 nanosecondes
Quartile 2 algo 1 : 3469 nanosecondes
Quartile 3 algo 1 : 3953 nanosecondes

* ***** *

Temps total algo 2 : 4146917 nanosecondes
Temps moyen algo 2 : 13823 nanosecondes
Temps max algo 2 : 68038 nanosecondes
Temps min algo 2 : 4741 nanosecondes
Variance algo 2 : 32467204 nanosecondes
Ecart type algo 2 : 5698 nanosecondes
Quartile 1 algo 2 : 9743 nanosecondes
Quartile 2 algo 2 : 11653 nanosecondes
Quartile 3 algo 2 : 14682 nanosecondes

Résultats & tests.

*** Résolutions de 300 Sudokus

Taille : N = 3 Indices = 28

Temps total algo 1 : 1211536908 nanosecondes
Temps moyen algo 1 : 4038456 nanosecondes
Temps max algo 1 : 61237263 nanosecondes
Temps min algo 1 : 52479 nanosecondes
Variance algo 1 : 7717945178161 nanosecondes
Ecart type algo 1 : 2778119 nanosecondes
Quartile 1 algo 1 : 751338 nanosecondes
Quartile 2 algo 1 : 1512490 nanosecondes
Quartile 3 algo 1 : 4928710 nanosecondes

* ***** *

Temps total algo 2 : 19836860 nanosecondes
Temps moyen algo 2 : 66122 nanosecondes
Temps max algo 2 : 2912342 nanosecondes
Temps min algo 2 : 29012 nanosecondes
Variance algo 2 : 186158736 nanosecondes
Ecart type algo 2 : 13644 nanosecondes
Quartile 1 algo 2 : 33706 nanosecondes
Quartile 2 algo 2 : 49065 nanosecondes
Quartile 3 algo 2 : 62292 nanosecondes

*** Résolutions de 300 Sudokus

Taille : N = 3 Indices = 35

Temps total algo 1 : 134917667 nanosecondes
Temps moyen algo 1 : 449725 nanosecondes
Temps max algo 1 : 26469966 nanosecondes
Temps min algo 1 : 17406 nanosecondes
Variance algo 1 : 176430241296 nanosecondes
Ecart type algo 1 : 420036 nanosecondes
Quartile 1 algo 1 : 56246 nanosecondes
Quartile 2 algo 1 : 125687 nanosecondes
Quartile 3 algo 1 : 304195 nanosecondes

* ***** *

Temps total algo 2 : 21329848 nanosecondes
Temps moyen algo 2 : 71099 nanosecondes
Temps max algo 2 : 14040904 nanosecondes
Temps min algo 2 : 11003 nanosecondes
Variance algo 2 : 3536799841 nanosecondes
Ecart type algo 2 : 59471 nanosecondes
Quartile 1 algo 2 : 12541 nanosecondes
Quartile 2 algo 2 : 15037 nanosecondes
Quartile 3 algo 2 : 19956 nanosecondes

Résultats & tests.

*** Résolutions de 300 Sudokus

Taille : N = 3 Indices = 35

Temps total algo 3 : 16223295 nanosecondes

Temps moyen algo 3 : 54077 nanosecondes

Temps max algo 3 : 254173 nanosecondes

Temps min algo 3 : 20860 nanosecondes

Variance algo 3 : 1000203876 nanosecondes

Ecart type algo 3 : 31626 nanosecondes

Quartile 1 algo 3 : 23415 nanosecondes

Quartile 2 algo 3 : 46836 nanosecondes

Quartile 3 algo 3 : 62408 nanosecondes

Résultats & tests.

*** Résolutions de 20 Sudokus

Taille : N = 4 Indices = 150

Temps total algo 1 : 85623967 nanosecondes
Temps moyen algo 1 : 4281198 nanosecondes
Temps max algo 1 : 36946674 nanosecondes
Temps min algo 1 : 207518 nanosecondes
Variance algo 1 : 15849674757225 nanosecondes
Ecart type algo 1 : 3981165 nanosecondes
Quartile 1 algo 1 : 452240 nanosecondes
Quartile 2 algo 1 : 1387465 nanosecondes
Quartile 3 algo 1 : 4303581 nanosecondes

* ***** *

Temps total algo 2 : 3212633 nanosecondes
Temps moyen algo 2 : 160631 nanosecondes
Temps max algo 2 : 942730 nanosecondes
Temps min algo 2 : 81211 nanosecondes
Variance algo 2 : 2427729984 nanosecondes
Ecart type algo 2 : 49272 nanosecondes
Quartile 1 algo 2 : 110563 nanosecondes
Quartile 2 algo 2 : 116040 nanosecondes
Quartile 3 algo 2 : 124282 nanosecondes

*** Résolutions de 20 Sudokus

Taille : N = 4 Indices = 200

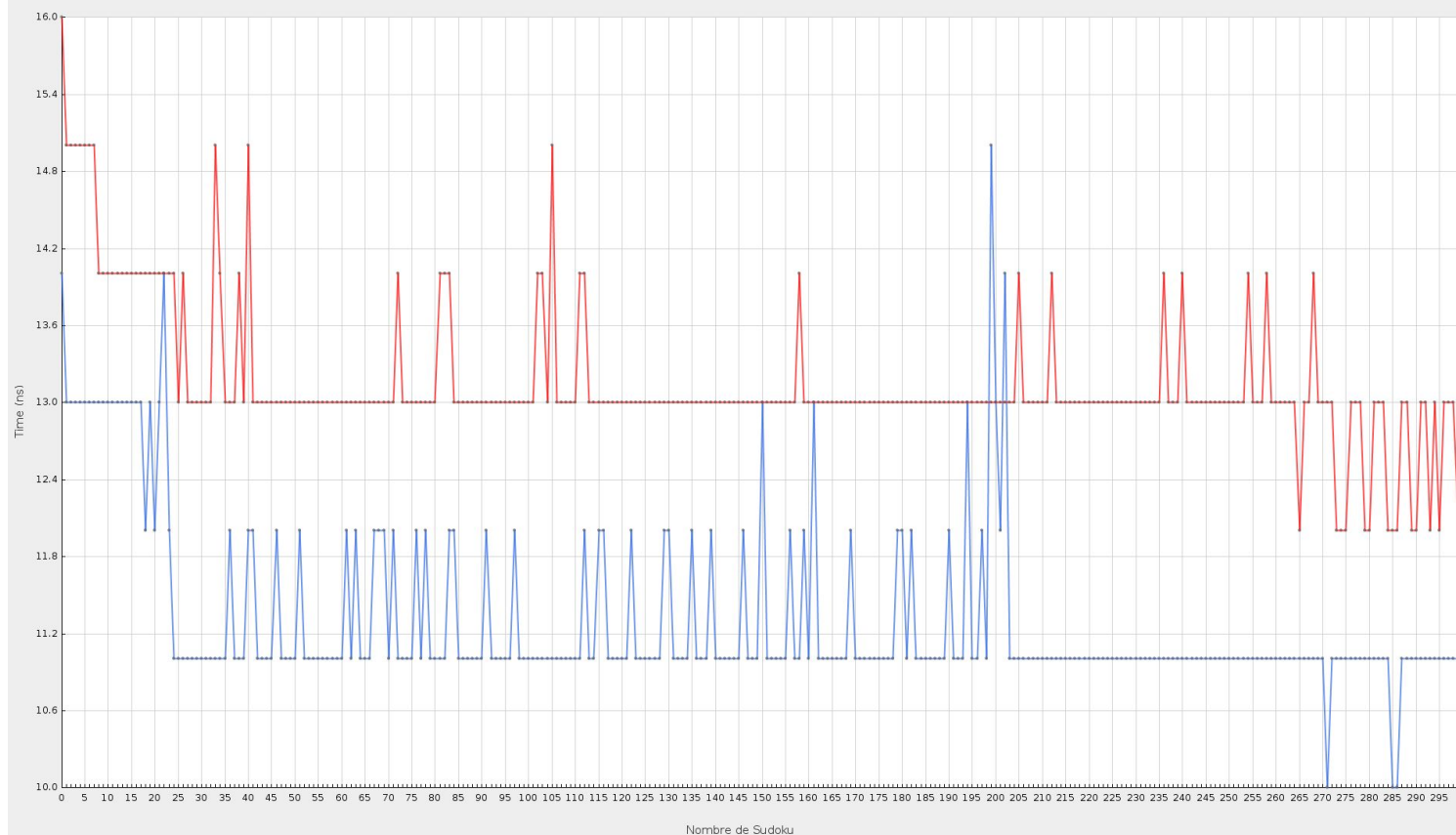
Temps total algo 1 : 17030384 nanosecondes
Temps moyen algo 1 : 851519 nanosecondes
Temps max algo 1 : 2019134 nanosecondes
Temps min algo 1 : 218658 nanosecondes
Variance algo 1 : 313215957649 nanosecondes
Ecart type algo 1 : 559657 nanosecondes
Quartile 1 algo 1 : 349843 nanosecondes
Quartile 2 algo 1 : 719948 nanosecondes
Quartile 3 algo 1 : 1271383 nanosecondes

* ***** *

Temps total algo 2 : 2646005 nanosecondes
Temps moyen algo 2 : 132300 nanosecondes
Temps max algo 2 : 380841 nanosecondes
Temps min algo 2 : 100966 nanosecondes
Variance algo 2 : 591511041 nanosecondes
Ecart type algo 2 : 24321 nanosecondes
Quartile 1 algo 2 : 107125 nanosecondes
Quartile 2 algo 2 : 115094 nanosecondes
Quartile 3 algo 2 : 135039 nanosecondes

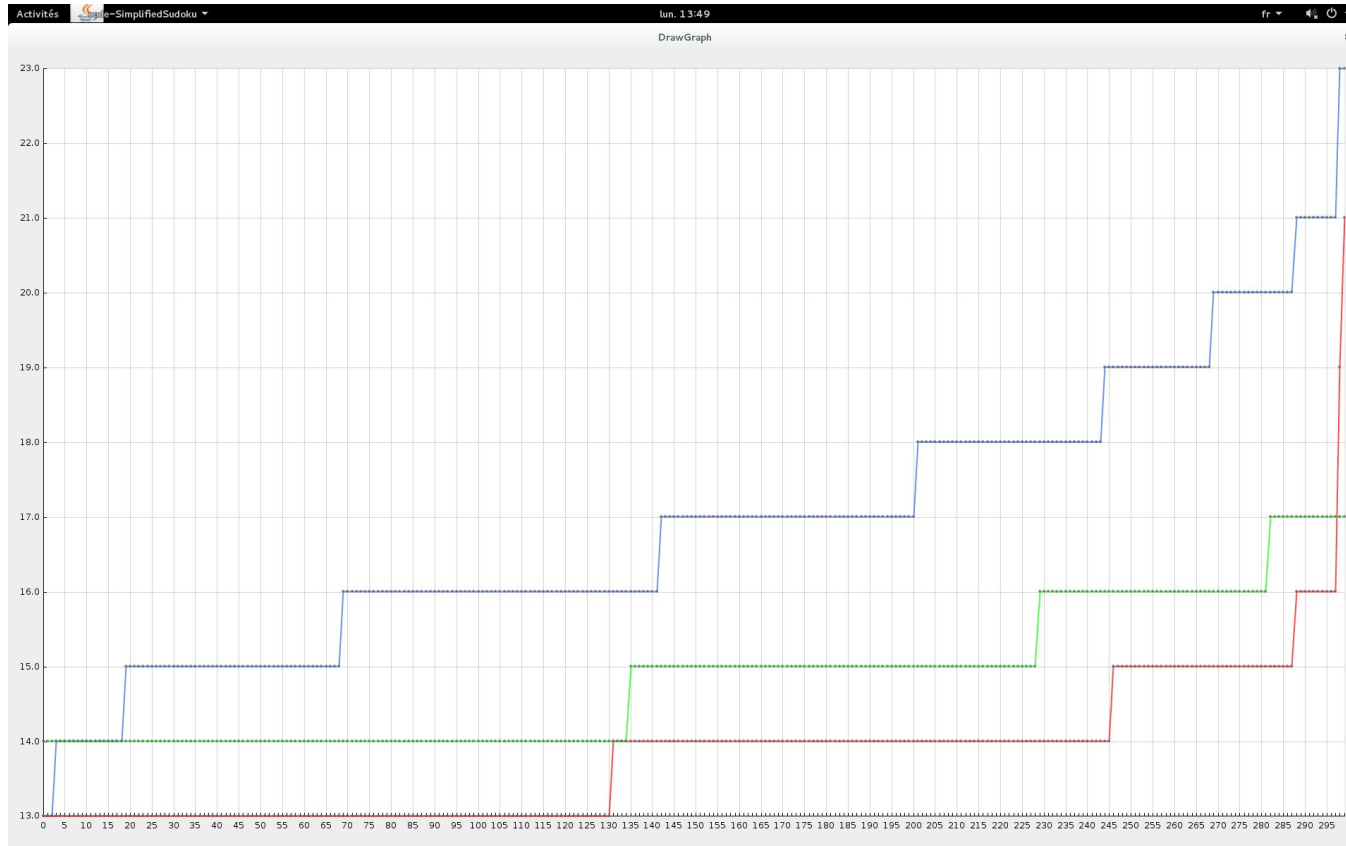
Courbe - Sudoku de taille $N = 2$ et Indices = 7

Temps de résolution sur un ensemble de Sudoku



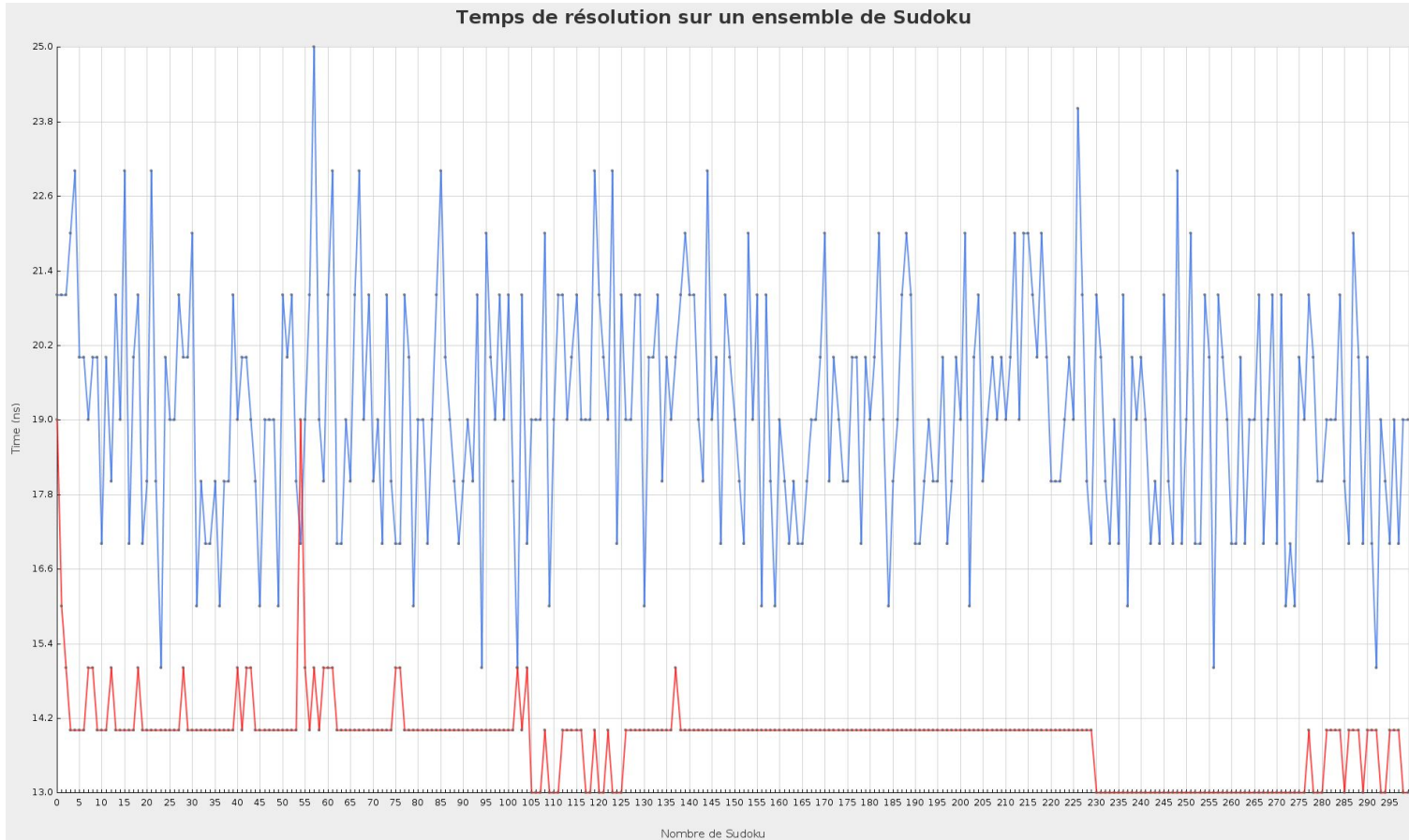
- Algorithme Naïf
- Algorithme Adaptatif

Courbe - Sudoku de taille $N = 3$ et Indices = 35



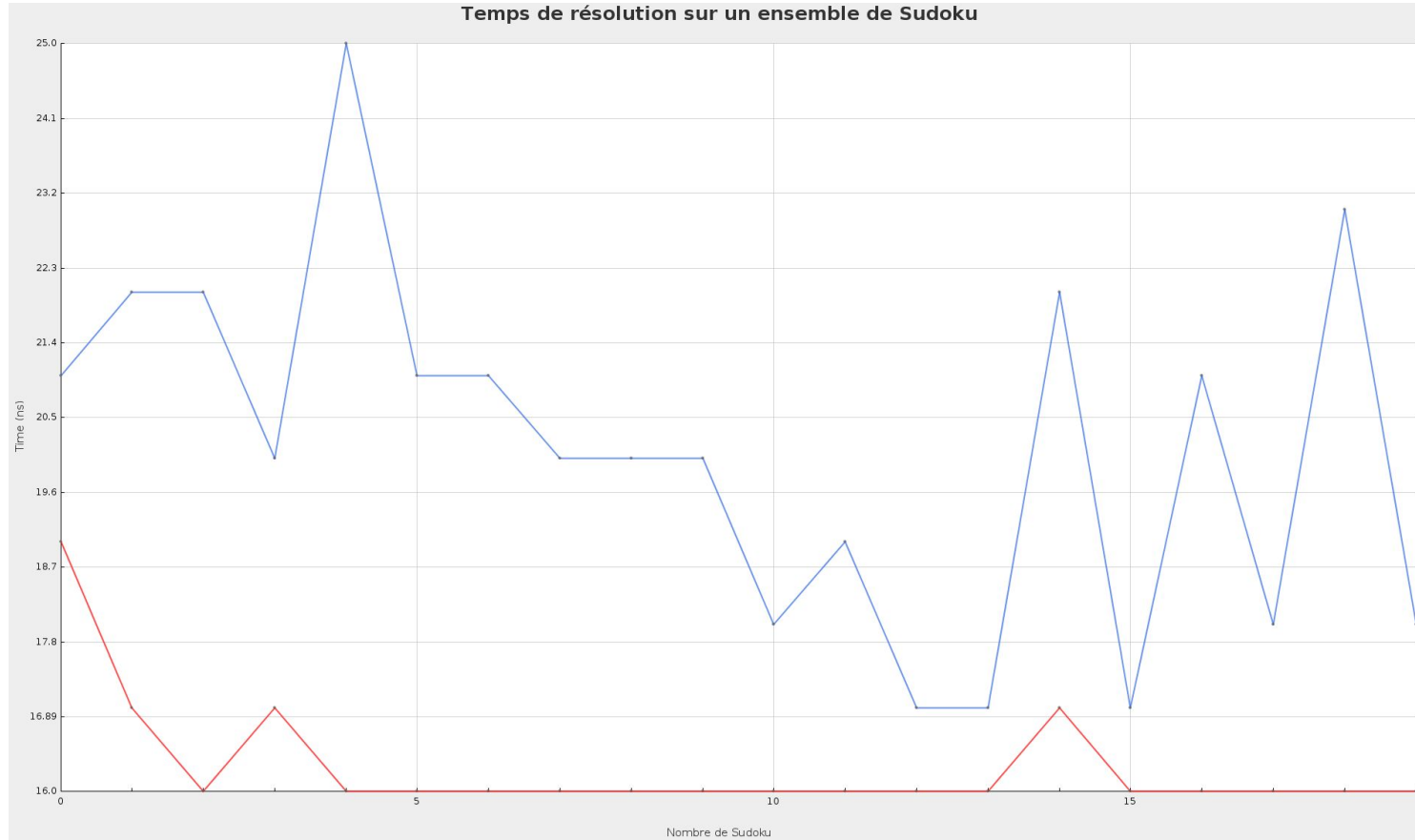
- Algorithme Naïf
- Algorithme Adaptatif

Courbe - Sudoku de taille $N = 3$ et Indices = 28



- Algorithme Naïf
- Algorithme Adaptatif

Courbe - Sudoku de taille $N = 4$ et Indices = 150



- Algorithme Naïf
- Algorithme Adaptatif

Algorithme adaptatif appliqué au Chaos Sudoku

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

Tableau du Sudoku:

```

{{3, 0, 0, 0, 0, 0, 0, 0, 4},
 {0, 0, 2, 0, 6, 0, 1, 0, 0},
 {0, 1, 0, 9, 0, 8, 0, 2, 0},
 {0, 0, 5, 0, 0, 0, 6, 0, 0},
 {0, 2, 0, 0, 0, 0, 0, 6, 0, 0},
 {0, 0, 9, 0, 0, 0, 8, 0, 0},
 {0, 8, 0, 3, 0, 4, 0, 6, 0},
 {0, 0, 4, 0, 1, 0, 9, 0, 0},
 {5, 0, 0, 0, 0, 0, 0, 0, 7}};
    
```

Tableau des régions:

```

{{1, 1, 1, 2, 3, 3, 3, 3, 3},
 {1, 1, 1, 2, 2, 2, 3, 3, 3},
 {1, 4, 4, 4, 4, 2, 2, 2, 3},
 {1, 1, 4, 5, 5, 5, 5, 2, 2},
 {4, 4, 4, 4, 5, 6, 6, 6, 6},
 {7, 7, 5, 5, 5, 5, 6, 9, 9},
 {8, 7, 7, 7, 6, 6, 6, 6, 9},
 {8, 8, 8, 7, 7, 7, 9, 9, 9},
 {8, 8, 8, 8, 8, 7, 9, 9, 9}};
    
```

Algorithme adaptatif appliqué au Chaos Sudoku

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

Tableau du Sudoku:

```
{{3, 0, 0, 0, 0, 0, 0, 0, 4},  
{0, 0, 2, 0, 6, 0, 1, 0, 0},  
{0, 1, 0, 9, 0, 8, 0, 2, 0},  
{0, 0, 5, 0, 0, 0, 6, 0, 0},  
{0, 2, 0, 0, 0, 0, 0, 0, 1},  
{0, 0, 9, 0, 0, 0, 8, 0, 0},  
{0, 8, 0, 3, 0, 4, 0, 6, 0},  
{0, 0, 4, 0, 1, 0, 9, 0, 0},  
{5, 0, 0, 0, 0, 0, 0, 0, 7}};
```

Tableau des régions:

```
{{1, 1, 1, 2, 3, 3, 3, 3, 3},  
{1, 1, 1, 2, 2, 2, 3, 3, 3},  
{1, 4, 4, 4, 4, 2, 2, 2, 3},  
{1, 1, 4, 5, 5, 5, 5, 2, 2},  
{4, 4, 4, 4, 5, 6, 6, 6, 6},  
{7, 7, 5, 5, 5, 5, 6, 9, 9},  
{8, 7, 7, 7, 6, 6, 6, 6, 9},  
{8, 8, 8, 7, 7, 7, 9, 9, 9},  
{8, 8, 8, 8, 8, 7, 9, 9, 9}};
```

protected boolean checkBox(int[][] region, int row, int col, int num);

Algorithme adaptatif appliqué au Chaos Sudoku

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

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

Résultat:

3 5 8 1 9 6 2 7 4
4 9 2 5 6 7 1 3 8
6 1 3 9 7 8 4 2 5
1 7 5 8 4 2 6 9 3
8 2 6 4 5 3 7 1 9
2 4 9 7 3 0 8 5 6
9 8 7 3 2 4 5 6 1
7 3 4 6 1 5 9 8 2
5 6 1 2 8 9 3 4 7

305097920 nanosecondes

Questions ?

Merci !
Pour nous contacter:

david.toty@etu.upmc.fr

maxime.tran@etu.upmc.fr

Encadrant du projet:

Responsable: Fabrice KORDON

Client: Tarek Menouer
