Présentation 6
PSAR

David TOTY

Maxime TRAN

Objectif:
Modéliser et résoudre le problème du Sudoku

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:

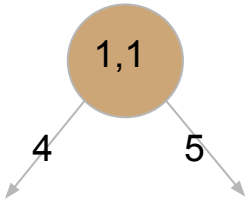
Variables: Les 81 cases du Sudoku
Domaines: L'ensemble {1, 2, 3, 4, 5, 6, 7, 8, 9}
Contraintes: 1 seule occurrence de valeur sur chaque ligne, colonne et région

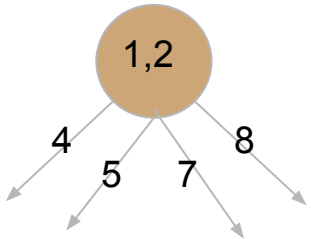
Le sudoku.

		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		

Comment résoudre un sudoku à la main ?

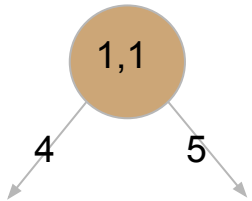
		3		2		6		
9			3		5			1
		1	8		6	4		
		8	1		2	9		
7								8
		6	7		8	2		
		2	6		9	5		
8			2		3			9
		5		1		3		

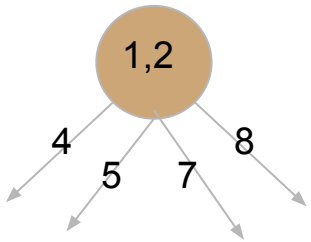


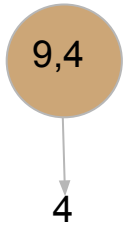


Comment résoudre un sudoku à la main ?

		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	X	1		3		







Á ce jour, 3 solveurs.

Solve(int row, int col);

Solve2(int row, int col,int xx);

Solve3();

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 poids:

5 3 4 5 2 3 2 4 4
5 3 3 5 3 3 3 4 4
6 3 4 4 2 3 2 3 4
4 4 5 4 3 2 4 1 4
4 2 4 3 3 3 5 2 4
3 2 4 3 3 3 4 3 3
3 4 3 4 1 5 4 3 5
4 4 3 4 2 5 3 2 5
4 4 4 2 1 4 3 3 5

Tableau de parcours: (i, x, y, poids)

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

Algorithme adaptatif

```
public void initPoids();

public void initParcours();

public void initFait();
```

Tableaux:

- Possibilité
- Etages

Algorithme adaptatif

public void initPoids();

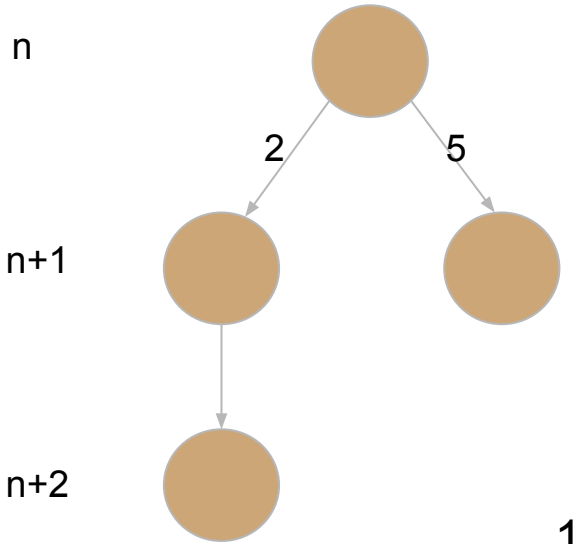
public void initParcours();

public void initFait();

Tableaux:
- Possibilité
- Etages

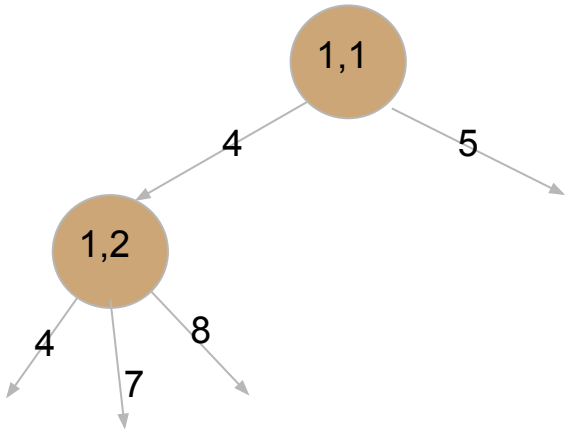
Tableau etages stock n.

Tableau de possibilités stock la valeur 5.



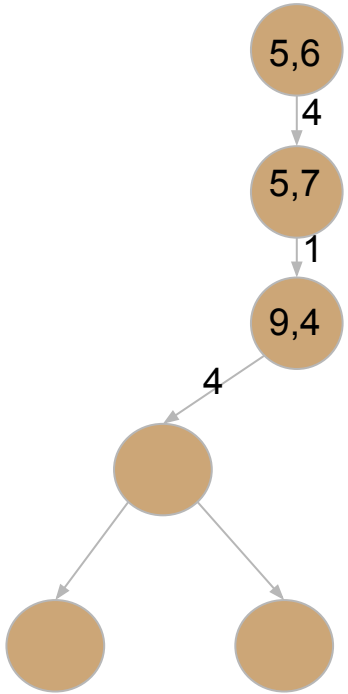
Comparaison de parcours de solution

		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		



Comparaison de parcours de solution

		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		



Calculs de poids dynamique.

```
public void solve3();

 void TrouverNext(int row, int col);
```

Tableaux:

- NextX
- NextY

On trouve les n cases de poids le plus faible, on assigne ces cases, on recommence.

Calculs de poids dynamique.

```
public void solve3();

 void TrouverNext(int row, int col);
```

Tableaux:
- NextX
- NextY

On trouve les n cases de poids le plus faible, on assigne ces cases, on recommence.

Ex :

3 : 7 : 1
6 : 4 : 1
8 : 4 : 1
0 : 4 : 2
0 : 6 : 2
2 : 4 : 2
...

- NextX = [3, 6, 8]
- NextY = [7, 4, 4]

Résultats & tests.

*** Resolutions de 300 Sudokus
Taille : 9

Temps total algo 1 : 320381224 nanosecondes
Temps moyen algo 1 : 1067937 nanosecondes
Temps max algo 1 : 27291677 nanosecondes
Temps min algo 1 : 19153 nanosecondes
Variance algo 1 : 687028792384 nanosecondes
Ecart type algo 1 : 828872 nanosecondes
Quartile 1 algo 1 : 133661 nanosecondes
Quartile 2 algo 1 : 355006 nanosecondes
Quartile 3 algo 1 : 924272 nanosecondes

* ********** *

Temps total algo 2 : 16633799 nanosecondes
Temps moyen algo 2 : 55445 nanosecondes
Temps max algo 2 : 3742578 nanosecondes
Temps min algo 2 : 22939 nanosecondes
Variance algo 2 : 958212025 nanosecondes
Ecart type algo 2 : 30955 nanosecondes
Quartile 1 algo 2 : 24743 nanosecondes
Quartile 2 algo 2 : 26155 nanosecondes
Quartile 3 algo 2 : 35968 nanosecondes

Résultats & tests.

*** Resolutions de 300 Sudokus
Taille : 4

Temps total algo 1 : 548593 nanosecondes
Temps moyen algo 1 : 1828 nanosecondes
Temps max algo 1 : 19917 nanosecondes
Temps min algo 1 : 892 nanosecondes
Variance algo 1 : 670761 nanosecondes
Ecart type algo 1 : 819 nanosecondes
Quartile 1 algo 1 : 1018 nanosecondes
Quartile 2 algo 1 : 1211 nanosecondes
Quartile 3 algo 1 : 1311 nanosecondes

* ********** *

Temps total algo 2 : 8968034 nanosecondes
Temps moyen algo 2 : 29893 nanosecondes
Temps max algo 2 : 2871259 nanosecondes
Temps min algo 2 : 8918 nanosecondes
Variance algo 2 : 432972864 nanosecondes
Ecart type algo 2 : 20808 nanosecondes
Quartile 1 algo 2 : 10652 nanosecondes
Quartile 2 algo 2 : 18810 nanosecondes
Quartile 3 algo 2 : 27206 nanosecondes

Résultats & tests.

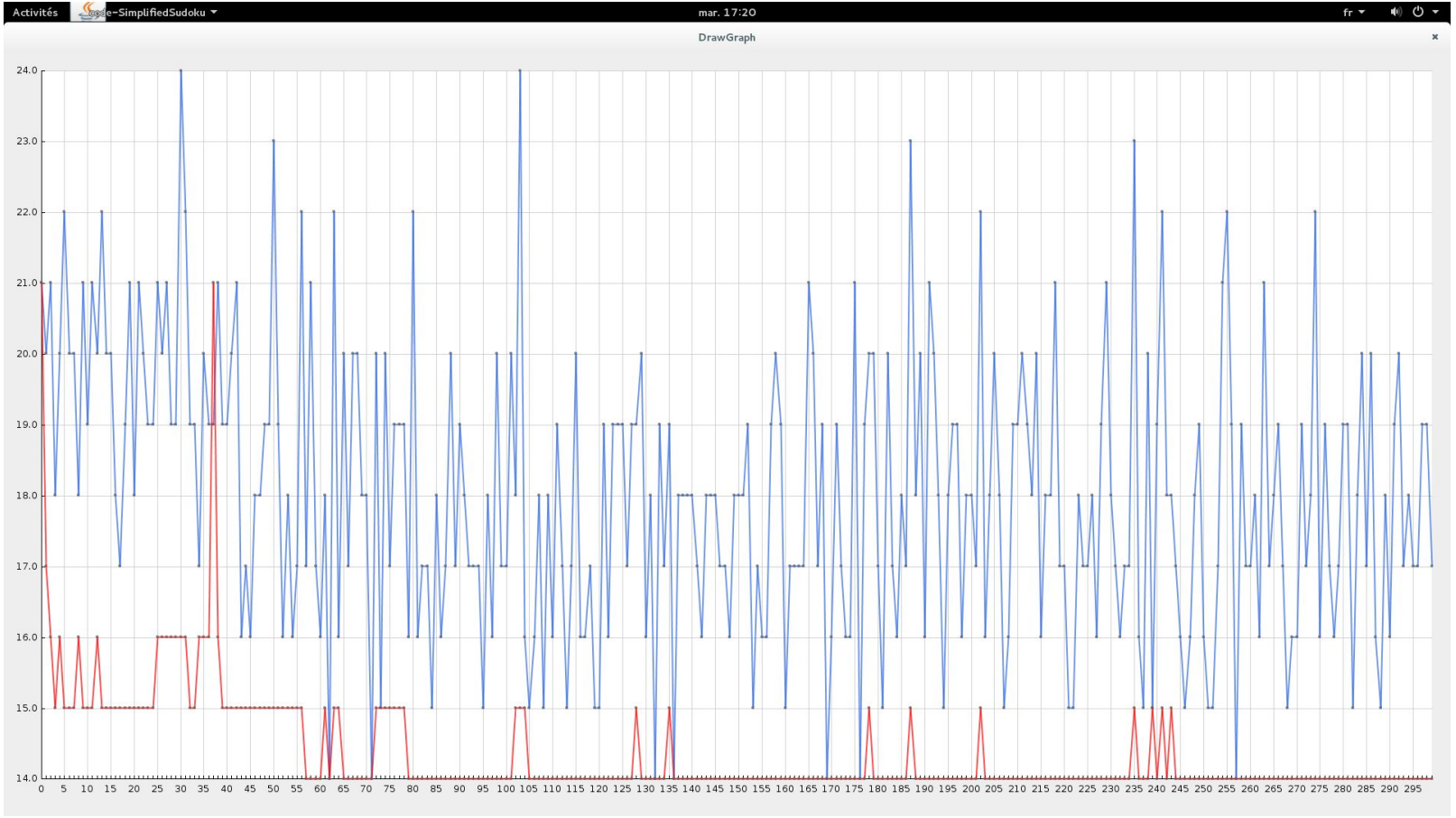
*** Resolutions de 20 Sudokus
Taille : 16

Temps total algo 1 : 126773906 nanosecondes
Temps moyen algo 1 : 6338695 nanosecondes
Temps max algo 1 : 51210051 nanosecondes
Temps min algo 1 : 237791 nanosecondes
Variance algo 1 : 34900222184881 nanosecondes
Ecart type algo 1 : 5907641 nanosecondes
Quartile 1 algo 1 : 1155061 nanosecondes
Quartile 2 algo 1 : 1882490 nanosecondes
Quartile 3 algo 1 : 8046959 nanosecondes

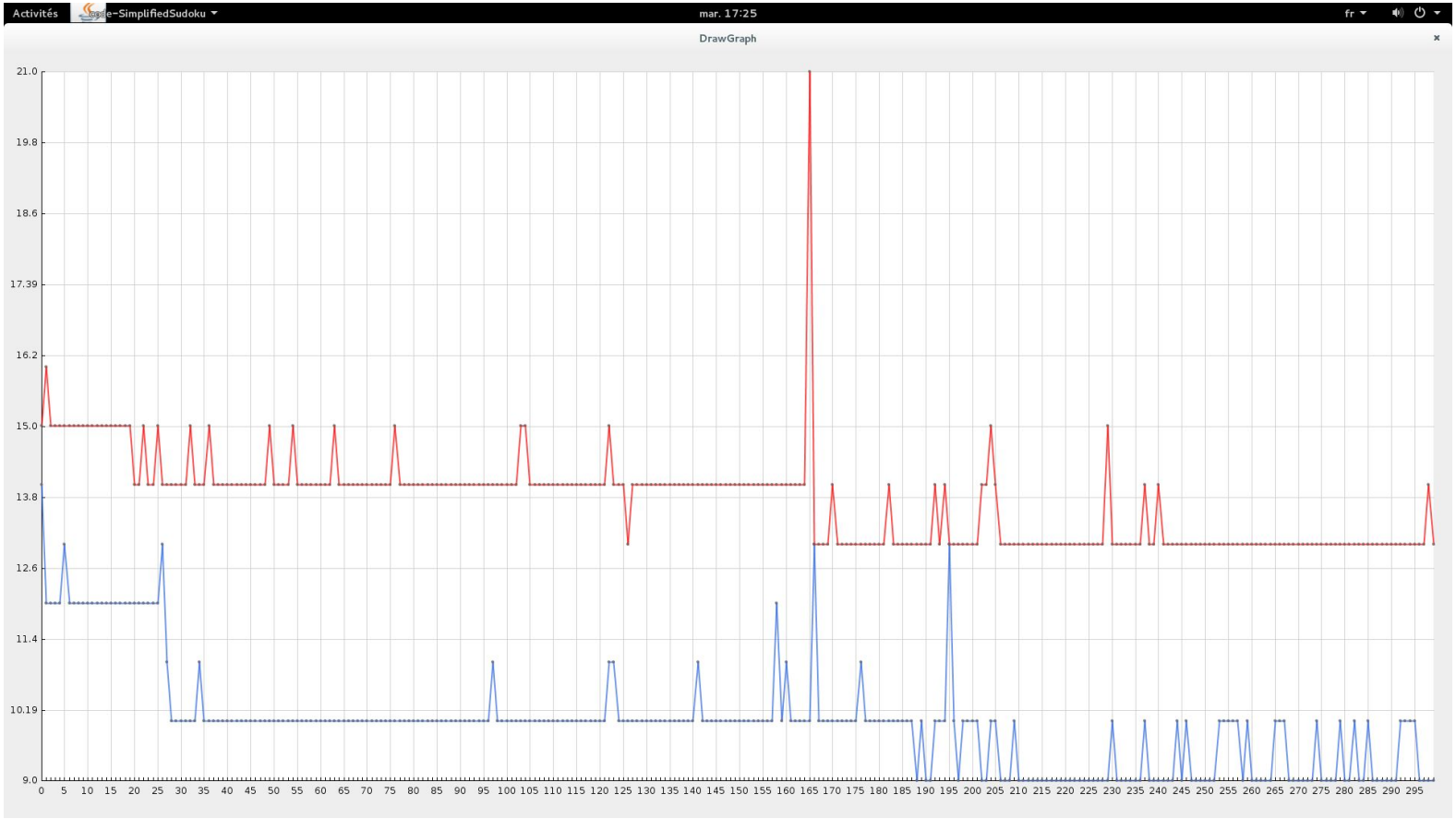
* ********** *

Temps total algo 2 : 6642703 nanosecondes
Temps moyen algo 2 : 332135 nanosecondes
Temps max algo 2 : 3525263 nanosecondes
Temps min algo 2 : 121116 nanosecondes
Variance algo 2 : 44529018361 nanosecondes
Ecart type algo 2 : 211019 nanosecondes
Quartile 1 algo 2 : 129178 nanosecondes
Quartile 2 algo 2 : 161365 nanosecondes
Quartile 3 algo 2 : 190925 nanosecondes

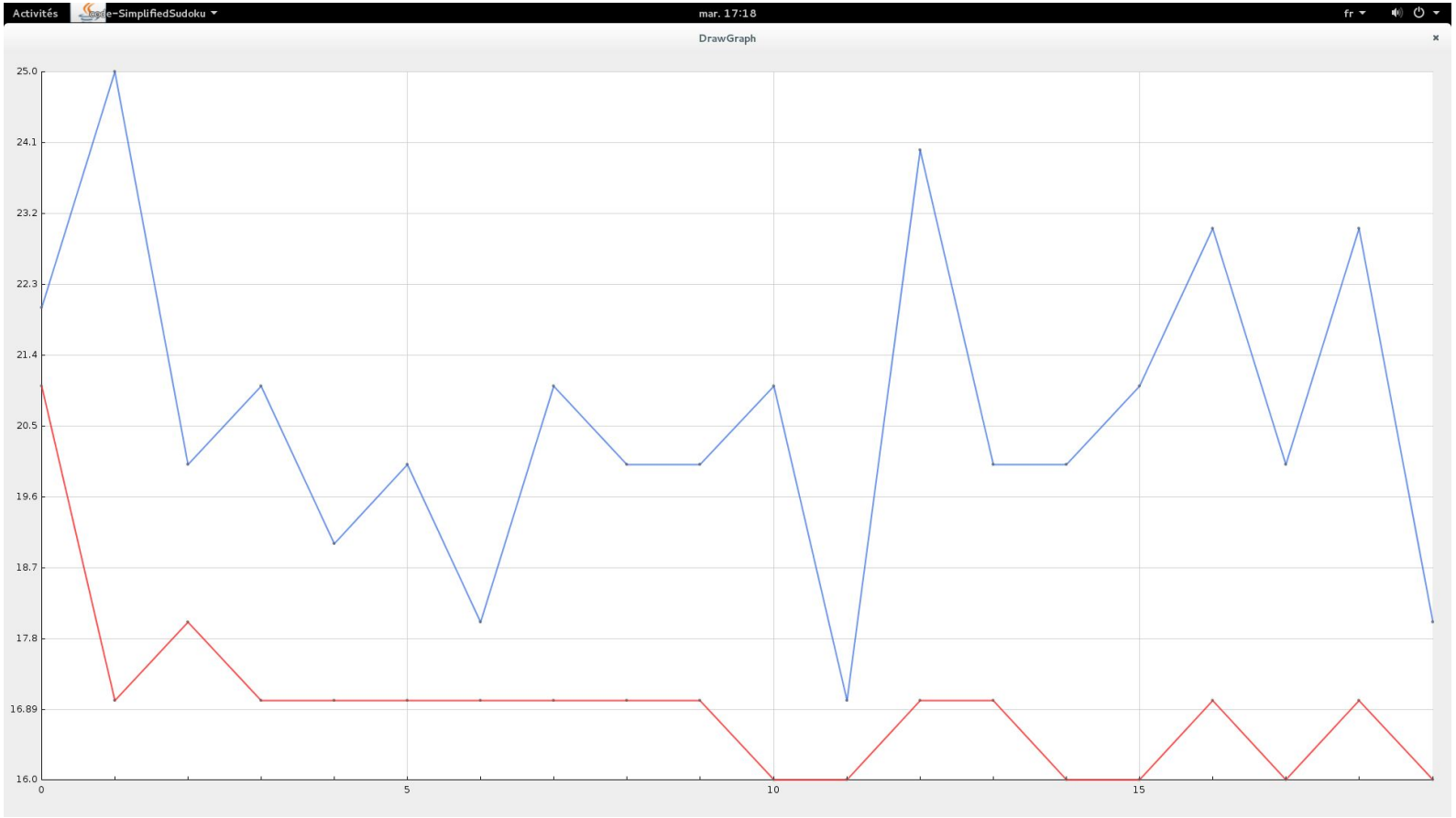
Courbes taille 9



Courbes taille 4



Courbes taille 16



Conclusion