Le problème des N dames
ou une première approche des heuristiques

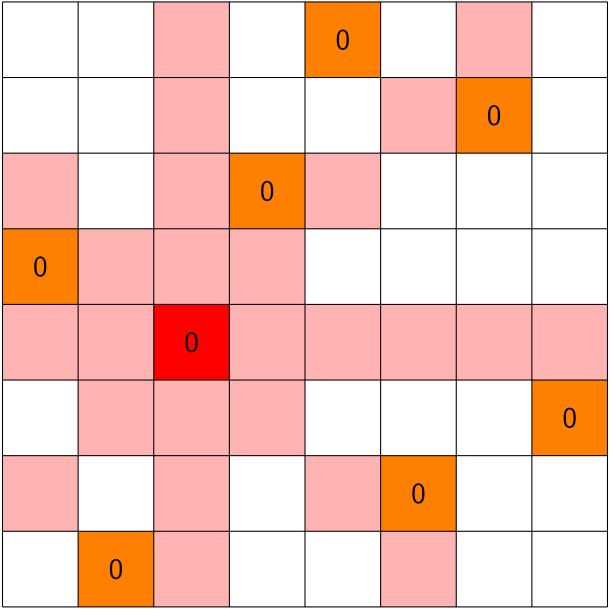
José Vander Meulen

16 novembre 2017

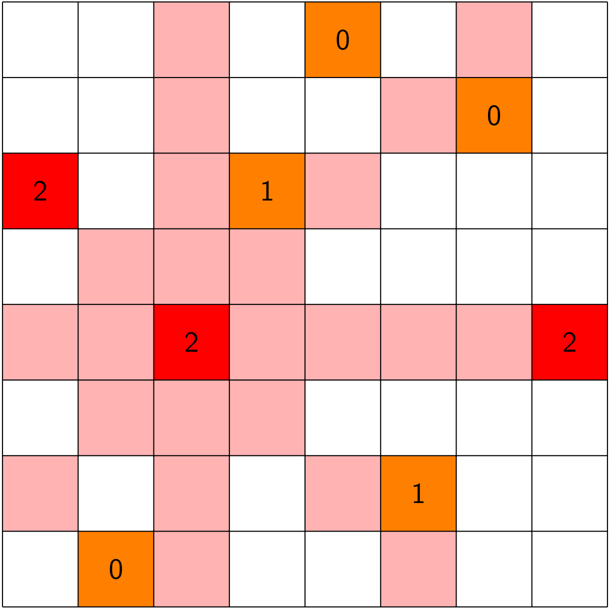
Problème

				0			
						0	
			0				
0							
		0					
							0
					0		
	0						

Problème



Problème



Min 2

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

Algorithme

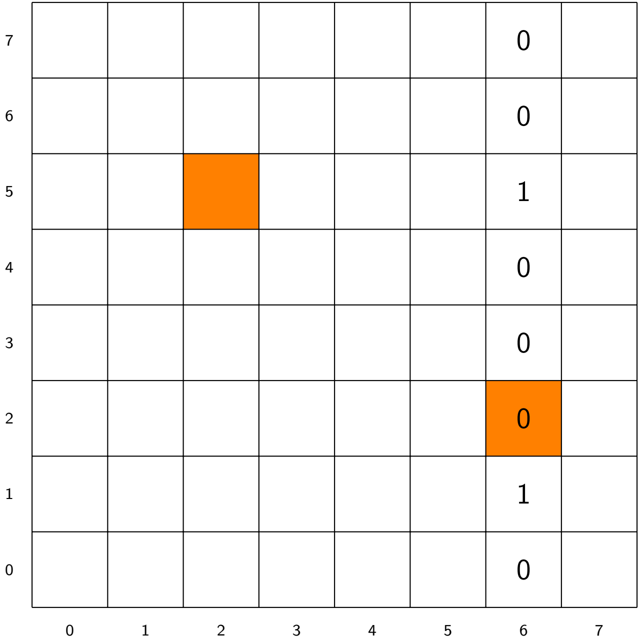
On procède en deux phases :

- On génère un première version (+/- bonne)
- Successivement, on essaye d'améliorer la solution actuelle

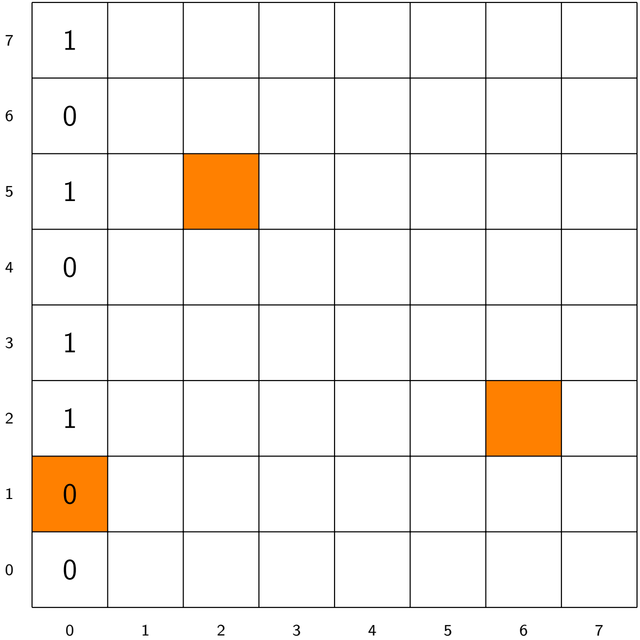
Première Phase : Une première solution

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

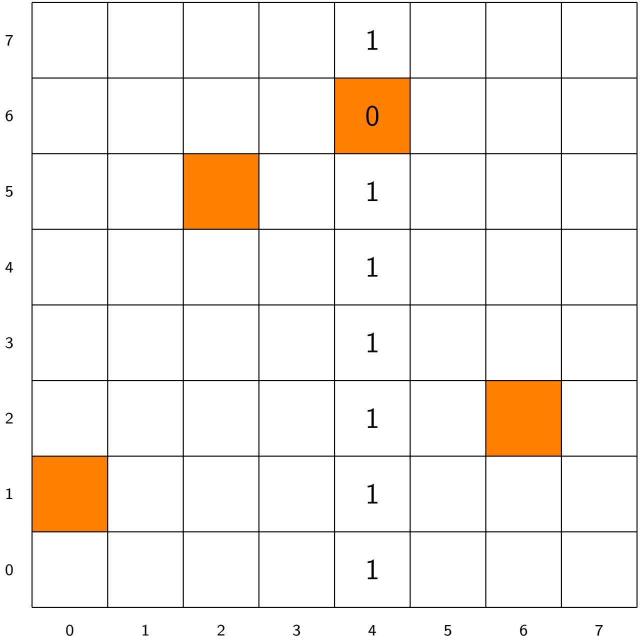
Première Phase : Une première solution



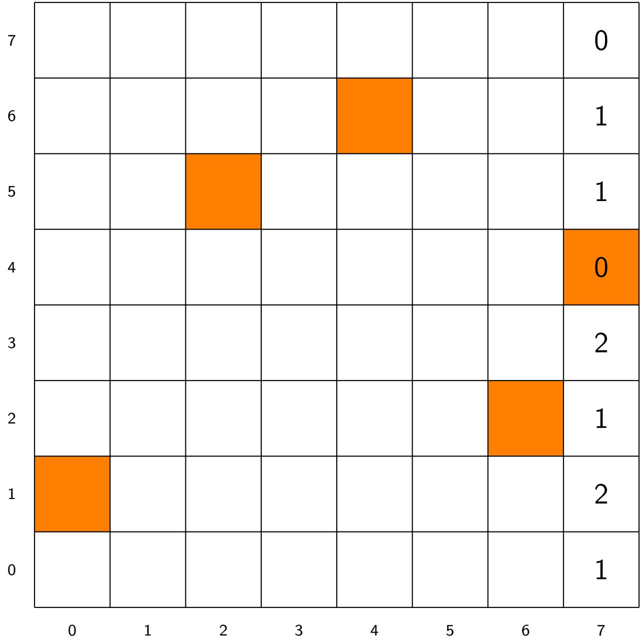
Première Phase : Une première solution



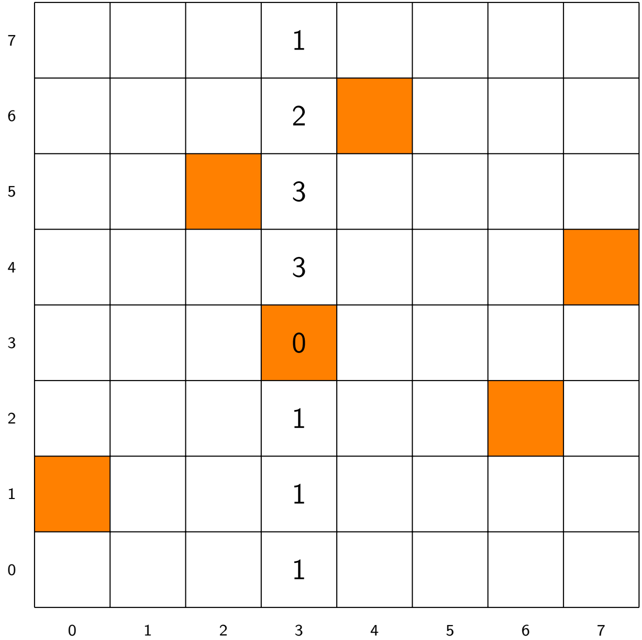
Première Phase : Une première solution



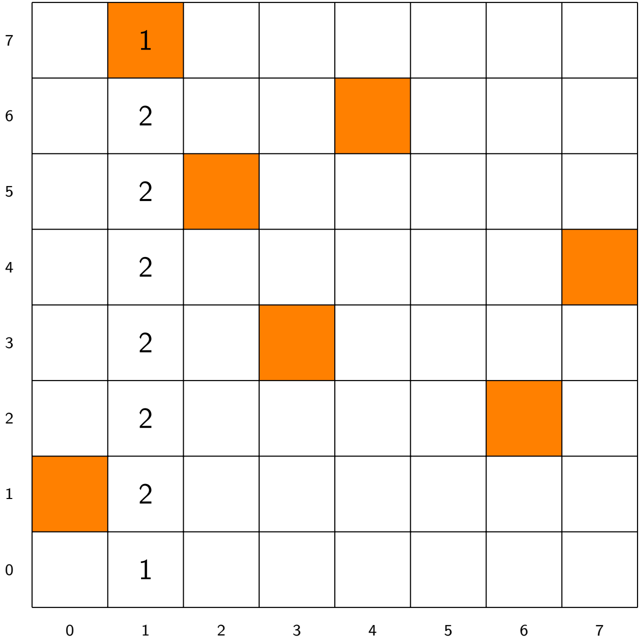
Première Phase : Une première solution



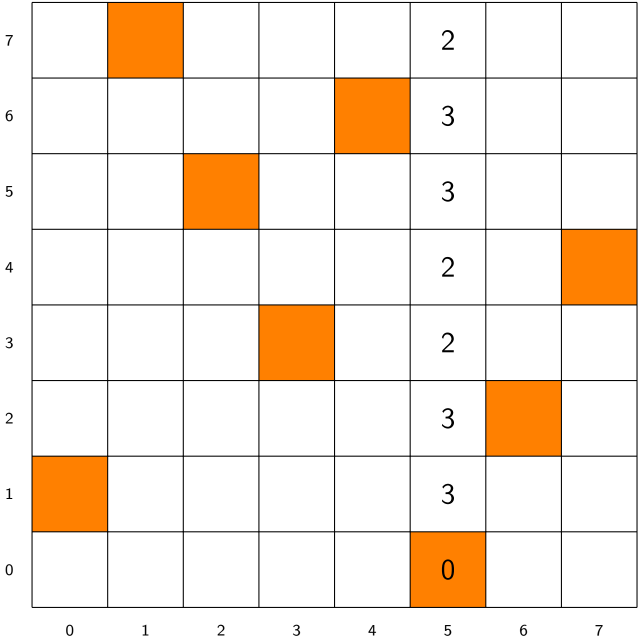
Première Phase : Une première solution



Première Phase : Une première solution



Première Phase : Une première solution



Première Phase : Une première solution

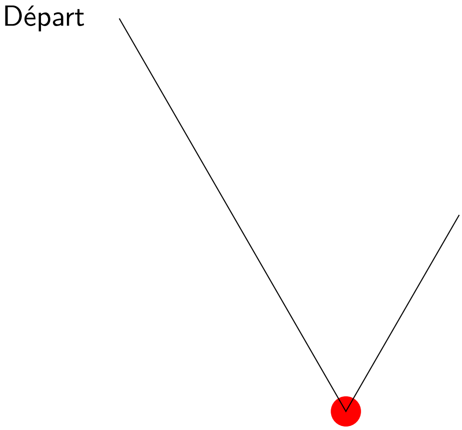
7		1						
6					0			
5			0					
4								0
3				0				
2							1	
1	0							
0						0		
	0	1	2	3	4	5	6	7

Seconde Phase : On améliore la solution courante

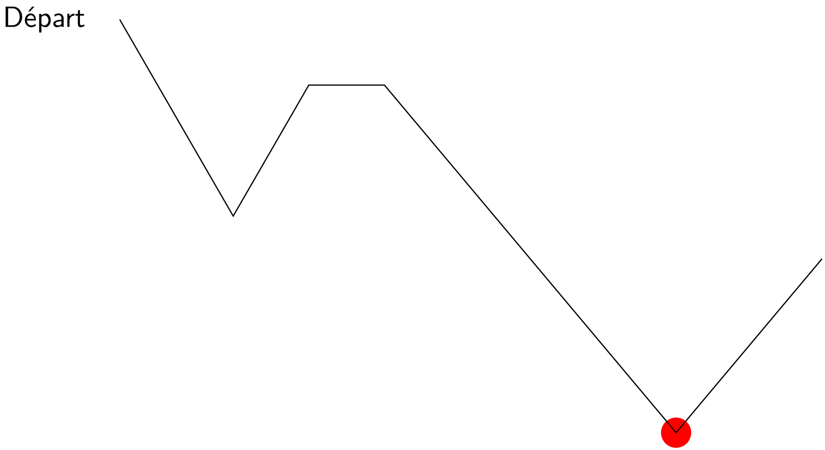
Tant qu'il reste des conflits :

- On choisit aléatoirement une des reines R qui est la plus en conflit
- On déplace la reine R de telle manière à minimiser ses conflits. S'il existe plusieurs cases qui minimisent ses conflits, on en choisit une aléatoirement

Minimum global



Minimum local

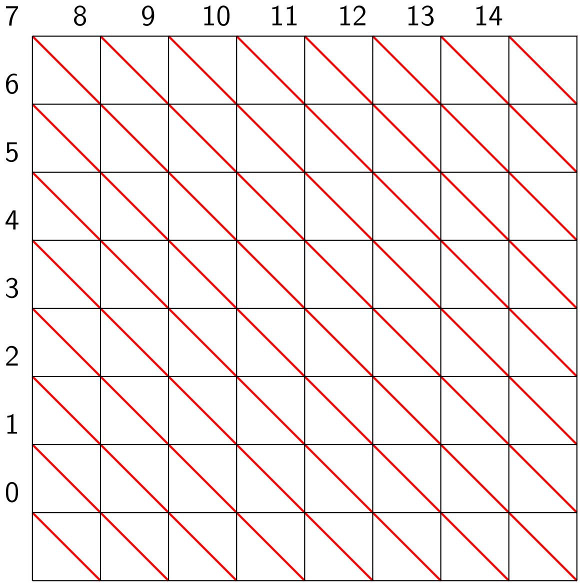


Seconde Phase 2.0 : Pour sortir des minimums locaux

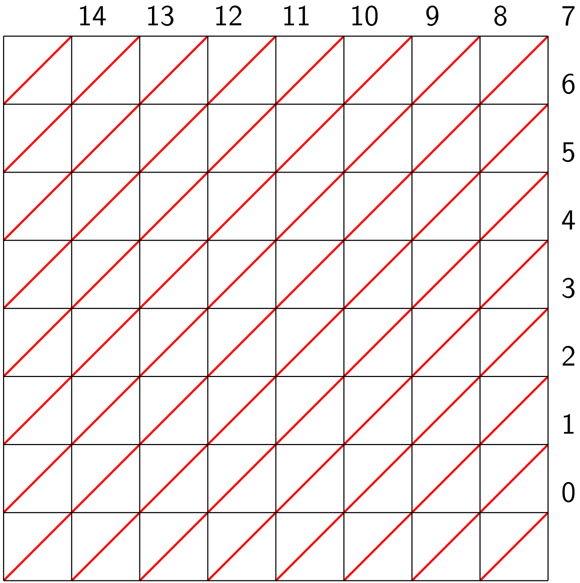
Tant qu'il reste des conflits :

- Dans $\approx 10\%$ des cas :
  - On choisit aléatoirement une des reines et sur une case choisie aléatoirement
- Dans $\approx 90\%$ des cas :
  - On choisit aléatoirement une des reines R qui est la plus en conflit
  - On déplace la reine R de telle manière à minimiser ses conflits. S'il existe plusieurs cases qui minimise ses conflits, on en choisit une aléatoirement

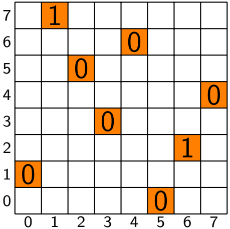
Numéro des diagonales descantes (i + j)



Numéro des diagonales montantes (N - 1 + j - i)



Structure de données



plateau

1	7	5	3	6	0	2	4

lignes

1	1	1	1	1	1	1	1

diagonales descendantes

0	1	0	0	0	1	1	1	2	0	1	1	0	0	0

diagonales montantes

0	0	1	1	1	0	0	1	1	1	1	0	0	1	0

Terminaison de l'algorithme

Théoriquement, il est possible que l'algorithme ne se termine pas.

Pratiquement, il se termine tjr lorsque $N = 0 \vee N = 1 \vee 4 \leq N \leq 20000$