

PROJET PPC

OLIVIER BAILLEUX

UNIVERSITÉ DE BOURGOGNE - MASTER INFORMATIQUE 2^{ÈME} ANNÉE BDIA



Rédigé en L^AT_EX par KÉVIN GRILLET

1 Principe

L'objectif est de réaliser une pré-génération, ce développement s'est découpé en deux phases, un développement de découverte et mise en place des fonctionnalités, puis une refonte du code pour optimiser l'algorithme.

1.1 Phase 1 - Mise en place des fonctionnalités

Dans un premier temps, implémentation d'un tirage au hasard k fois en vérifiant simplement si la case n'était pas déjà utilisée. Suite à cela, ajout de la vérification des diagonales.

Afin d'éviter que l'algorithme tourne indéfiniment, ajout d'une limite du nombre de tentative avant que le générateur ne s'arrête.

Répétition jusqu'à ce que k soit atteint :

Tirage au sort d'un X et d'un Y .

Vérification des diagonales.

Si pas de problème lors de la vérification des diagonales :

Enregistrement de X et Y dans `init`.

Remise à zéro du nombre de tentatives.

Sinon :

Incrémentation du nombre de tentatives.

Si nombre de tentatives trop grand :

Arrêt du générateur.

1.2 Phase 2 - Optimisation

Dans un second temps, ajout des listes contenant les lignes et colonnes n'ayant pas encore de reines et ainsi éviter une grosse redondance dans le tirage de la reine à placer. Changement de la vérification des diagonales pour rendre le tout plus compact et s'arrêtant plus rapidement en cas de conflit.

Ces modifications ont permis d'améliorer les performances sans toucher au nombre de tentatives avant l'arrêt du générateur.

Génération des listes des lignes et colonnes possibles dans pX et pY .

Répétition jusqu'à ce que k soit atteint :

Tirage au sort d'un X et d'un Y dans pX , pY .

Vérification des diagonales.

Si pas de problème lors de la vérification des diagonales :

Enregistrement de X et Y dans `init`.

Suppression de X et Y dans pX et pY .

Remise à zéro du nombre de tentatives.

Sinon :

Incrémentation du nombre de tentatives.

Si nombre de tentative trop grand :

Arrêt du générateur.

2 Structures de données

`int[] init` Structure fonctionnant comme `IntVar[] vars` de choco, stoquant les X et Y de la manière suivante : `init[x] → y + 1`.

`ArrayList<Integer> pX, pY` Structure stockant les lignes et colonnes sans reines. Les X vont de 0 à $n - 1$ et les Y vont de 1 à n pour correspondre aux structures `init` et `vars`.

`int[][] diag` Structure stockant les 4 directions des diagonales, permettant de vérifier toutes les directions avec une simple boucle.

3 Algorithmes

ALGORITHME generate

// Générateur maison qui place k reines sur les n

VARIABLES

```
chk, isValid : BOOLEAN;
iTry, x, y, dx, dy, cpt, nb : INTEGER;
init : INTEGER[];
pX, pY : ARRAYLIST<INTEGER>;
```

DEBUT

```
pX <- gpX; //Liste des X (0..n-1)
pY <- gpY; //Liste des X (1..n)
init <- new INTEGER[n];
iTry <- 0;
nb <- 0;
TANT QUE (nb < k) FAIRE
  x <- Tirage random de pX;
  y <- Tirage random de pY;
  chk <- VRAI;
  cpt <- 1;
  // Vérification des diagonales, temps qu'on a pas d'erreur
  TANT QUE (chk ET isValid ET nb > 0) FAIRE
    chk <- FAUX;
    // On boucle sur les 4 directions
    POUR i ALLANT DE 0 A 4 FAIRE
      dx <- diag[i][0] * cpt;
      dy <- diag[i][1] * cpt;
      // Vérification globale pour savoir si on est dans la grille
      SI (x + dx >= 0 ET x + dx < n ET y + dy > 0 ET y + dy < n + 1) ALORS
        chk <- VRAI;
        SI () ALORS
          isValid <- FAUX;
          STOP;
        FIN SI
      FIN SI
    FIN POUR
    ctp <- cpt + 1;
  FIN TANT QUE
  SI isValid ALORS
    init[x] <- y;
    // On retire le X/Y utilisé
    pX.REMOVE(x);
    pY.REMOVE(y);
  SINON
    SI (iTry > n * 1000) ALORS
      ARRET;
    SINON
      iTry <- iTry + 1;
    FIN SI
  FIN SI
FIN TANT QUE
```

FIN

4 Résultats expérimentaux

real K	n																															
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Total général	
1	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	
2		1,00	1,92	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	1,96
3			1,87	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	2,96
4				3,23	3,88	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00	3,97
5					4,31	4,85	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	4,97
6						4,95	5,65	5,99	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	5,94
7							5,76	6,48	6,93	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	7,00	6,92
8								6,57	7,43	7,93	7,99	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	7,91
9									7,50	8,29	8,91	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	9,00	8,90
10										8,33	9,11	9,80	9,99	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	9,87
11											9,22	10,12	10,79	10,97	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	11,00	10,86
12												10,09	10,99	11,74	11,97	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	12,00	11,83
13													11,14	11,99	12,69	12,96	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	13,00	12,82
14														11,84	12,94	13,68	13,96	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	13,79
15															12,80	13,81	14,64	14,95	14,99	15,00	15,00	15,00	15,00	15,00	15,00	15,00	15,00	15,00	15,00	15,00	15,00	14,76
16																13,79	14,69	15,61	15,96	16,00	16,00	16,00	16,00	16,00	16,00	16,00	16,00	16,00	16,00	16,00	16,00	15,74
17																	14,75	15,52	16,45	16,89	17,00	17,00	17,00	17,00	17,00	17,00	17,00	17,00	17,00	17,00	17,00	16,69
18																		15,62	16,53	17,51	17,89	18,00	18,00	18,00	18,00	18,00	18,00	18,00	18,00	18,00	18,00	17,66
19																			16,79	17,45	18,36	18,87	19,00	19,00	19,00	19,00	19,00	19,00	19,00	19,00	19,00	18,62
20																				17,45	18,47	19,29	19,88	20,00	20,00	20,00	20,00	20,00	20,00	20,00	20,00	19,55
21																					18,53	19,35	20,31	20,86	20,97	21,00	21,00	21,00	21,00	21,00	21,00	20,50
22																						19,54	20,44	21,25	21,76	21,97	22,00	22,00	22,00	22,00	22,00	21,44
23																							20,43	21,29	22,19	22,76	22,97	23,00	23,00	23,00	23,00	22,33
24																								21,29	22,19	23,14	23,74	23,95	24,00	24,00	24,00	23,19
25																									22,09	23,13	23,97	24,78	24,96	25,00	25,00	23,99
26																										23,12	24,17	24,85	25,69	26,00	26,00	24,77
27																											23,91	24,95	25,87	26,46	26,46	25,30
28																												24,99	26,02	26,93	26,93	25,98
29																													25,97	26,84	26,84	26,41
30																														26,82	26,82	26,82
Total général	1,00	1,00	1,60	2,31	2,84	3,30	3,77	4,26	4,76	5,26	5,75	6,25	6,76	7,25	7,76	8,27	8,77	9,26	9,77	10,27	10,77	11,28	11,79	12,28	12,77	13,27	13,77	14,27	14,78	15,27	10,43	

On peut voir que l'algorithme, pour $N = 30$, arrive à placer toutes les reines jusqu'à $K = N - 4$, entre $K = N - 4$ et $K = N - 3$ il manque 1 reine. Plus K tend vers N , plus le placement d'une reine au hasard est difficile. Plus N est grand, plus le problème apparaît tôt.

result	n																																
k		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Total général	
1		100	0	0	48	100	62	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	2710	
2			0	0	26	71	18	88	78	99	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	2480
3				0	17	51	8	37	37	60	73	94	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	2277
4					23	44	2	22	14	28	20	55	82	99	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	2089
5						38	8	8	13	14	11	20	25	69	85	98	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	1889
6							5	6	7	14	5	2	12	19	48	77	92	99	100	100	100	100	100	100	100	100	100	100	100	100	100	100	1686
7								12	6	7	5	7	7	9	17	39	67	90	99	100	100	100	100	100	100	100	100	100	100	100	100	100	1565
8									3	5	0	3	2	3	7	11	22	47	83	98	100	100	100	100	100	100	100	100	100	100	100	100	1384
9										3	0	3	0	2	3	3	8	20	36	69	87	99	100	100	100	100	100	100	100	100	100	100	1233
10											5	0	1	2	0	1	5	8	19	23	52	74	98	100	100	100	100	100	100	100	100	100	1088
11												2	0	2	0	1	3	3	3	7	23	35	64	91	100	100	100	100	100	100	100	100	934
12													2	1	0	0	3	0	3	5	7	12	38	52	86	98	99	100	100	100	100	100	806
13														3	0	2	1	0	2	2	2	3	12	30	49	75	97	100	100	100	100	100	678
14															0	1	0	2	0	1	3	4	5	12	26	37	73	94	100	100	100	100	558
15																2	1	0	0	1	0	2	1	2	10	16	36	61	79	96	100	100	407
16																	1	1	0	0	0	1	0	1	0	5	7	10	23	45	76	90	260
17																		0	0	0	0	0	1	2	0	1	1	7	5	26	49	70	162
18																			0	0	0	0	0	1	1	1	1	2	5	9	15	34	69
19																				0	1	0	0	0	2	1	0	1	0	4	10	11	30
20																					0	0	1	0	0	0	0	0	1	1	1	6	10
21																						0	0	0	0	0	0	0	1	1	1	2	5
22																							0	0	0	0	0	0	1	0	2	2	5
23																								0	0	0	0	0	0	1	0	0	1
24																									0	0	0	0	0	0	0	0	0
25																										0	0	0	0	0	0	0	0
26																											0	0	0	0	0	0	0
27																												0	1	0	0	0	1
28																													0	0	1	0	1
29																														0	0	1	1
30																															1	0	1
Total général		100	0	0	114	304	103	273	258	330	319	386	431	509	560	635	703	770	845	907	975	1030	1123	1190	1279	1335	1425	1492	1566	1652	1716	22330	

backtrack	n																																	
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Total général			
1	0,00	0,00	0,00	0,93	0,12	5,71	1,06	4,10	4,91	13,62	20,84	21,58	16,98	13,67	25,78	25,00	22,44	37,30	26,86	28,32	18,38	22,54	16,88	20,67	20,40	19,24	18,59	29,87	20,81	17,27	15,80			
2		0,00	0,00	0,00	0,02	1,07	0,98	3,81	4,97	13,08	9,37	14,30	17,57	16,12	20,87	25,53	20,25	22,75	22,63	23,37	19,17	20,64	38,58	16,37	20,05	15,85	13,85	28,45	39,58	18,39	15,44			
3			0,00	0,00	0,00	0,07	0,23	0,82	2,72	7,84	12,68	11,91	12,14	17,66	17,11	12,77	14,52	16,51	14,18	17,19	21,30	19,81	22,41	12,65	14,36	16,53	14,56	23,73	15,55	16,79	12,00			
4				0,00	0,00	0,03	0,00	0,08	0,28	2,31	4,48	10,10	12,56	13,08	14,14	15,14	12,60	18,87	17,43	12,37	26,24	14,68	23,95	18,11	13,43	18,37	14,25	19,17	14,54	21,81	11,78			
5					0,00	0,00	0,00	0,03	0,03	0,04	0,69	3,79	6,77	11,03	14,00	14,69	18,54	12,92	11,91	24,31	17,22	14,48	12,08	13,48	13,27	15,97	16,04	21,99	17,82	17,21	10,70			
6						0,00	0,00	0,00	0,00	0,06	0,06	0,42	2,17	4,63	8,94	13,06	20,29	11,57	13,76	10,28	12,61	13,63	12,46	11,19	17,73	12,33	12,69	22,19	18,86	42,52	10,46			
7							0,00	0,00	0,00	0,03	0,03	0,09	0,28	1,34	3,03	6,27	12,24	11,33	10,64	11,28	10,85	12,07	13,38	12,40	11,24	8,25	15,47	14,23	13,90	19,34	7,82			
8								0,00	0,00	0,00	0,00	0,00	0,00	0,16	0,78	2,31	5,17	7,41	12,71	12,36	13,48	15,89	13,98	10,47	13,54	13,65	16,39	14,81	8,25	9,37	7,42			
9									0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,06	0,47	1,38	3,53	8,04	12,32	12,58	16,59	13,66	9,59	11,27	16,55	12,94	11,30	12,93	16,89	7,28		
10										0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,04	0,29	1,04	2,28	6,49	15,01	15,89	13,85	12,82	19,90	9,90	12,30	10,55	13,17	11,56	6,91		
11											0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,12	0,16	0,63	1,42	4,92	9,27	13,43	14,96	18,88	15,35	12,81	11,50	13,63	11,47	6,43		
12												0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	0,17	0,37	1,24	4,14	8,60	11,71	14,30	13,50	13,27	9,83	11,80	16,66	5,56		
13													0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,07	0,11	0,89	3,12	7,12	12,28	13,61	14,83	13,97	15,14	13,69	5,27		
14														0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,10	0,77	1,71	4,05	8,06	11,22	12,27	12,25	11,41	3,64		
15															0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,06	0,00	0,36	1,30	3,91	9,94	11,03	16,46	14,81	3,62		
16																0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,17	0,30	0,93	3,46	6,68	14,95	14,58	2,74		
17																	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	0,48	1,03	1,90	7,24	10,18	1,49	
18																		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,08	0,23	0,94	1,20	3,37	0,45	
19																			0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,06	0,28	1,47	0,15	
20																				0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,05	0,00	0,09	0,28	0,04	
21																					0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	
22																						0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
23																							0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
24																								0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
25																									0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
26																										0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
27																											0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
28																												0,00	0,00	0,00	0,00	0,00	0,00	0,00
29																													0,00	0,00	0,00	0,00	0,00	0,00
30																														0,00	0,00	0,00	0,00	0,00
Total général	0,00	0,00	0,00	0,23	0,03	1,15	0,32	1,11	1,43	3,70	4,38	5,18	5,27	5,55	6,98	7,21	7,52	7,97	7,43	8,01	8,24	8,21	9,01	7,24	8,25	7,79	7,92	9,45	9,26	9,64	7,09			

Globalement, le nombre de backtrack est bien inférieur lorsque l'on place des points. Ce nombre trouve un minima vers $K = N/2$. Le solveur dit 0 lorsque celui-ci ne trouve pas de solution.

5 Sources

L'intégralité de mon code source et mes résultats allant jusqu'à $N = 80$ (fichier de sortie et images) sont disponibles sur mon GitHub : <https://git.io/vNtHi>