

BUT Informatique

Rapport S2.02



Le problème des huit reines

**LAMBERT-BRUNEL Baptiste /BRUNEAU-GACHE Léo /
BUREL Dylan/POTIER Maëlan**

Année Universitaire 2023-2024

Sommaire

Introduction	2
Force brute	2
Backtracking	2
Graphe	3
Conclusion	3
Annexe	4

Introduction

Le problème consiste à placer sur un échiquier le maximum de reines possible sans qu'elles ne se menacent. Pour rappel une reine peut en menacer une autre sur sa ligne, colonne ou encore diagonale. Afin de résoudre ce problème, nous avons élaboré plusieurs algorithmes permettant de trouver les solutions sur plusieurs échiquiers de tailles différentes, ainsi de comparer notre approche sur ce problème. Ainsi nous avons fait l'algorithme dit de "force brute", le backtracking et la création d'un graphe.

Force brute

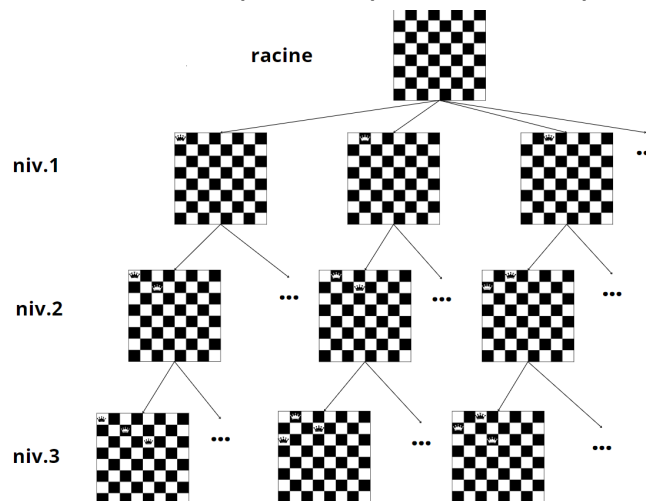
Pour résoudre ce problème, le premier algorithme que nous avons pensé est le fait de penser à tester toutes les possibilités sur un échiquier, c'est-à-dire de parcourir à chaque fois le jeu en entier. Cette solution assez rapide sur les petites grilles s'avère être compliquée voir impossible sur les plus grandes (voir annexe). Nous commençons par une position donnée puis en fonction d'elle nous parcourons la grille. La technique s'avère simple, dès qu'une reine est posée on essaye de trouver une autre solution en regardant l'entièreté de la grille de jeu. Cependant, c'est sur ce point que l'algorithme devient exponentiel avec un temps dépassant certaines limites. Néanmoins l'algorithme est assez simple et assez facile à conceptualiser. Au vu des exemples donnés nous avons donc vu qu'il était impossible de poursuivre dans cette voie nous avons donc choisi un nouvel algorithme basé sur une méthode bien connue.

Backtracking

La technique du backtracking est assez connue du grand public, elle consiste à prendre une case et essayer des possibilités prédéfinies avant. Ainsi elle nous permet de résoudre des grilles plus rapidement que la précédente technique. Le principe est simple, nous prenons en paramètre un échiquier vide, la ligne actuelle ou nous cherchons à ajouter une reine et puis une case ou nous posons la première reine. Ainsi nous pourrions parcourir toutes les cases de la ligne actuelle qui évoluera à chaque itération de notre programme. Si la case est possible et si le backtracking fonctionne à la case d'après nous posons une nouvelle reine. Dans le cas contraire, la case sera sans reine. Ainsi nous avons trouvé une méthode pour une résolution optimale. L'avantage de cet algorithme est sa rapidité pour trouver une seule solution pour une case donnée. Cependant, l'inconvénient majeur que nous avons relevé est sa complexité qui devient exponentielle. Nous pouvons donc passer à la dernière technique.

Graphe

La méthode du graphe consiste à représenter les différents états du jeu sur chacun de ses niveaux. Nous partons donc d'un jeu d'échecs vide, nous posons une reine sur la première case représentant le premier nœud. Ainsi de suite nous avons sur le premier niveau les 8 possibilités représentant les 8 états possibles sur la première ligne. Finalement, nous avons construit un graphe représentant l'entière des possibilités pour n'importe quelle taille d'échiquier. Ce graphe peut s'appeler arbre de décision. Pour illustrer cela nous pouvons prendre l'exemple d'un échiquier 8 * 8.



L'inconvénient principal que nous avons rencontré est le fait de représenter, de penser à des nœuds qui sont des échiquier et non des cases. Ce remplissage a été une étape indispensable pour travailler dessus et trouver un certain nombre de solutions par la suite. L'avantage de cet algorithme nous a permis de savoir (pour n'importe quelle case de n'importe quelle taille d'échiquier) le nombre de solutions par niveaux donc par ligne d'échiquier. De plus, bien que la création de l'arbre puisse prendre du temps, il suffit de le créer une seule fois pour ensuite pouvoir réaliser différentes opérations, qui sont très rapides.

Conclusion

Le problème des huit reines nous a permis de mettre en valeur nos compétences en programmation. Ainsi avec différents algorithmes, nous avons pu donner une approche à ce problème en donnant les différentes solutions que nous avons trouvées. De plus, ce travail nous a appris à gérer un projet de groupe avec l'utilisation d'un GitHub permettant de mettre en relation toutes nos ébauches.

Annexe

Taille de grille	Temps Backtracking (une solution reine imposée)	Temps Backtracking (nb solutions reine imposée)	Temps création graphe	Temps graphe (nb solutions)	Temps Graphe (nb solutions reine imposée)	Temps force brute (nb solutions reine imposée)
2	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
3	0.0 s	0.0 s	0.0s	0.0 s	0.00038 secondes	0.0 s
4	0.0 s	0.0 s	0.005358 s	0.0 s	0.00222 secondes	0.0 s
8	0.008005 s	0.00792 s	1.047156 s	0.0 s	1.14404 secondes	0.00199699 4 s
16	0.8689 s	3370.2 s (56 min)	*	*	*	792.782866 45 s (13 minutes)

*après avoir laissé tourner le programme environs 2 heures, il s'est brusquement arrêté avec un message d'erreur, c'est pourquoi les données sont manquantes

lien d'accès au diaporama :

https://www.canva.com/design/DAF-uYWIYwo/qtl2A23tbg87kVVfFwdfTA/edit?utm_content=DAF-uYWIYwo&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton