Introduction

Le problème des 8 reines est un problème mathématique classique qui consiste à placer 8 reines sur un échiquier de taille 8x8 de manière à ce qu'aucune reine ne puisse attaquer une autre reine.
Une reine peut attaquer une autre reine si elle se trouve sur la même ligne, colonne ou diagonale. Par conséquent, le problème consiste à trouver une disposition des 8 reines sur l'échiquier qui respecte cette contrainte.
Le problème des 8 reines est intéressant car il a des implications importantes dans de nombreux domaines, tels que la cryptographie, l'informatique théorique et l'intelligence artificielle.
Il existe de nombreuses approches pour résoudre le problème des 8 reines, notamment des algorithmes de recherche exhaustive, des algorithmes heuristiques et des méthodes de backtracking. Cependant, même avec des méthodes de résolution avancées, le problème des 8 reines peut être difficile à résoudre en raison du grand nombre de combinaisons possibles.

Voici quelques difficultés et particularités du problème des 8 reines :

- Bien que le problème ne semble pas trop difficile au premier abord, il est en réalité assez complexe. Il existe en effet 92 solutions différentes pour placer les 8 reines sur l'échiquier, et trouver la solution optimale nécessite de tester un grand nombre de configurations différentes.

- Le problème des 8 reines est un problème combinatoire, c'est-à-dire qu'il repose sur la recherche de toutes les combinaisons possibles. Pour résoudre le problème, il est donc nécessaire d'explorer toutes les possibilités.

- Si l'on trouve une solution sur un coin de l'échiquier, il est possible de la faire pivoter et l'appliquer sur les autres coins de l'échiquier. Cela signifie qu'il existe en réalité seulement 12 solutions uniques, et non 92.

- La méthode de recherche la plus couramment utilisée pour résoudre le problème des 8 reines est l'algorithme de backtracking. Cette méthode consiste à tester toutes les combinaisons possibles, en revenant en arrière chaque fois qu'une solution est incorrecte, jusqu'à trouver la solution optimale.

Pseudo code + Explications détaillées du problème des huit reines

```
fonction huit_reines(colonne):
    si colonne est en dehors de l'échiquier alors:
        retourner VRAI

    pour chaque ligne de l'échiquier faire:
        si la reine peut être placée en (ligne, colonne) sans être menacée alors:
            placer la reine en (ligne, colonne)
            si huit_reines(colonne + 1) retourne VRAI alors:
                retourner VRAI
            enlever la reine de (ligne, colonne)

    retourner FAUX
```

L'algorithme prend en entrée une colonne, qui représente la colonne de l'échiquier à partir de laquelle on doit placer les reines. Si la colonne est en dehors de l'échiquier (c'est-à-dire qu'on a placé une reine sur chaque colonne), alors l'algorithme retourne vrai, indiquant que l'échiquier a été rempli avec succès avec huit reines.
Sinon, pour chaque ligne de l'échiquier, l'algorithme vérifie si une reine peut être placée en (ligne, colonne) sans être menacée par une autre reine déjà placée. Si c'est le cas, la reine est placée sur cette position, et l'algorithme appelle récursivement la fonction huit_reines avec la colonne suivante.

Si huit_reines retourne vrai (c'est-à-dire que toutes les reines ont été placées avec succès sur les colonnes suivantes), alors l'algorithme retourne vrai. Sinon, l'algorithme enlève la reine de la position (ligne, colonne) et essaie une autre ligne.
Si aucune reine ne peut être placée dans cette colonne sans être menacée par les reines déjà placées, l'algorithme retourne faux, indiquant qu'il n'y a pas de solution possible à partir de cette position de départ.

Description des types d'algorithmes

Pour ce projet nous avons réalisé un programme par personne, soit au total 3 programmes.

Code Rayane

L'algorithme utilisé est un algorithme récursif de backtracking. L'idée générale de cet algorithme est d'essayer toutes les possibilités pour placer les reines sur l'échiquier, en utilisant la récursion pour explorer toutes les configurations possibles.
À chaque étape, l'algorithme place une reine dans une nouvelle colonne et vérifie si elle est menacée par une reine déjà placée sur l'échiquier. Si la reine est menacée, l'algorithme

revient en arrière (backtrack) et essaie une autre position pour la reine précédente, et ainsi de suite jusqu'à ce qu'il trouve une solution ou qu'il ait épuisé toutes les possibilités. L'algorithme de backtracking est une méthode systématique et exhaustive pour résoudre des problèmes de recherche de toutes les solutions possibles, mais il peut être coûteux en temps et en mémoire pour des problèmes complexes.

Code Alexandre

L'algorithme utilisé dans ce code est un algorithme récursif de recherche de solution pour le problème des huit reines. L'algorithme utilise une approche de backtracking pour explorer toutes les combinaisons possibles pour placer les huit reines sur un échiquier de taille 8x8. La fonction trouver_reine() est appelée récursivement pour chaque colonne en explorant toutes les lignes possibles pour placer une reine qui ne menace pas les autres reines déjà placées. Si une configuration ne fonctionne pas, l'algorithme revient en arrière et explore une autre combinaison possible. L'algorithme s'arrête lorsque toutes les huit reines sont placées sur l'échiquier sans se menacer mutuellement.

Code ALAN

L'algorithme utilisé pour résoudre ce problème est une recherche avec récursivité. L'algorithme parcourt toutes les cases de la première ligne et essaie de placer une reine dans chaque case. Ensuite, il passe à la deuxième ligne et essaie de placer une reine dans chaque case qui ne crée pas de conflit avec les reines déjà placées.

Si une reine ne peut pas être placée sur la deuxième ligne sans créer de conflit, l'algorithme revient en arrière et essaie de placer une autre reine sur la première ligne. Si toutes les reines ont été placées sans conflit, l'algorithme renvoie True et la solution est trouvée.

Comparaison entre 2 algorithmes

Pour cette partie , nous avons décidé de comparer le code de Alexandre avec le code de Rayane .Les deux codes résolvent le problème des 8 reines, mais ils utilisent des approches différentes.

Le code d’Alexandre utilise une approche récursive pour placer une reine à la fois sur le plateau et vérifier si elle est attaquée par une autre reine déjà placée. S'il n'y a pas de conflits, la fonction continue à placer des reines sur le plateau jusqu'à ce qu'il soit rempli de 8 reines ou qu'il n'y ait plus de possibilités pour placer des reines. Si aucune solution n'est trouvée, il renvoie "pas de solution".

Le code de Rayane utilise une approche de backtracking pour résoudre le problème. Il génère toutes les combinaisons possibles de positions de reine sur le plateau en utilisant une liste de coordonnées (ligne, colonne) et vérifie s'il n'y a pas de conflit entre les reines. S'il y a un conflit, il revient en arrière et essaie une autre position. S'il n'y a pas de conflit, il ajoute la position de la reine à la liste de coordonnées et continue à essayer de placer d'autres reines.

Dans l'ensemble, les deux codes sont efficaces pour résoudre le problème des 8 reines, mais l'approche récursive du premier code peut être plus difficile à comprendre, tandis que l'approche de backtracking du deuxième code est plus intuitive et facile à comprendre.

Conclusion

En conclusion, le problème des 8 reines est un problème connu qui peut se résoudre de différentes manières. Nous avons utilisé le système de backtracking mais il en existe d'autres comme l'algorithme génétique : cette méthode consiste à créer une population initiale de solutions possibles, puis à les évaluer et à les sélectionner en fonction de leur "fitness" (adéquation à la solution). Les solutions sélectionnées sont alors croisées et mutées pour créer une nouvelle génération de solutions, qui est à nouveau évaluée et sélectionnée. La recherche locale : cette méthode consiste à prendre une solution initiale et à la modifier progressivement pour la rendre meilleure. Par exemple, on peut prendre une solution aléatoire et la modifier en déplaçant une dame à une position meilleure jusqu'à ce qu'aucune amélioration ne soit possible et aussi la programmation par contraintes (CP) : cette méthode consiste à modéliser le problème sous forme de contraintes, puis à utiliser un solveur de CP pour trouver une solution satisfaisant toutes les contraintes. Cette méthode est souvent utilisée pour résoudre des CSP plus complexes que le problème des 8 dames. De plus, on peut aussi le faire de manière plus ou moins détaillé par exemple nous avons fait une interface graphique pour l'un des programmes ou encore certains programmes sont plus rapides que les autres. Pour finir, le problème des 8 dames est non seulement un casse-tête intéressant à résoudre, mais il a également des applications dans divers domaines, tels que l'optimisation des horaires et des emplois du temps, la planification des ressources, l'ordonnancement de production, etc…

Questions

Déterminer le nombre de solutions pour un placement imposé de la première reine.

Si la position de la première reine est imposée, alors le nombre de solutions possibles dépendra de la position de cette première reine.

Pouvez-vous déterminer l'existence d'autres solutions (en partant d'autres positions) sans faire tourner votre programme

En utilisant cette méthode, on trouve qu'il y a exactement 4 solutions possibles pour le problème des 6 dames sur un échiquier de 6x6 cases. Les voici (en notant les positions des dames sous forme de coordonnées (ligne, colonne)) :

Solution 1 : (1, 3), (2, 1), (3, 6), (4, 4), (5, 2), (6, 5)

Solution 2 : (1, 4), (2, 6), (3, 1), (4, 3), (5, 5), (6, 2)

Solution 3 : (1, 2), (2, 4), (3, 6), (4, 1), (5, 3), (6, 5)

Solution 4 : (1, 5), (2, 3), (3, 1), (4, 6), (5, 4), (6, 2)

On peut vérifier que chacune de ces solutions satisfait les contraintes du problème des 6 dames, c'est-à-dire qu'aucune dame n'est menacée par une autre dame, et qu'il y a exactement 6 dames sur l'échiquier.

Pouvez-vous déterminer toutes les solutions pour des échiquiers de 6 × 6 et 8 × 8. Qu'en est-il s'il l'on considère des échiquiers de taille quelconque n × n ?

Solutions pour nxn avec le temps d'exécution pour un PC lambda.

n	nombre de solutions	temps d'exécution
4	2	< 1 s
5	10	< 1 s
6	4	< 1 s
7	40	< 1 s
8	92	≈ 10 s
9	352	≈ 2 min
10	724	≈ 20 min
11	2680	≈ 3 h

A partir de n = 12, les calculs deviennent trop long mais certaines personnes ont pu aller jusqu'à n = 26.

Le nombre de possibilités pour un echiquier n*n reste donc encore flou, certains mathématiciens se sont penchés dessus main