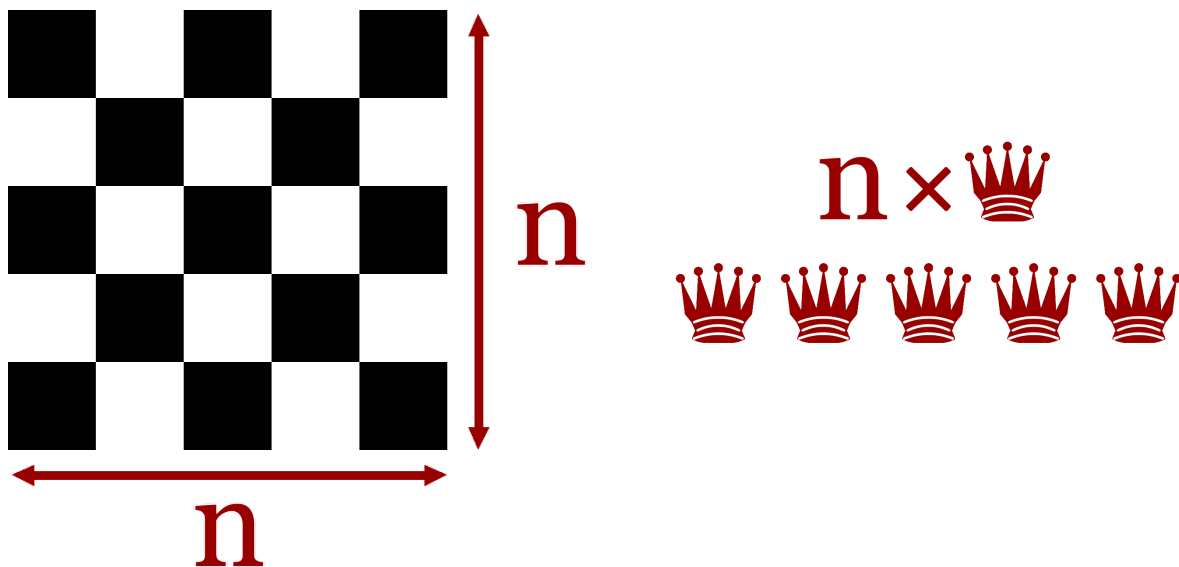


S2.02 : Exploration algorithmique

Problème des n reines

Introduction

Le problème des n reines est un problème mathématique classique qui consiste à placer n reines sur un échiquier de $n \times n$ cases, de manière à ce qu'aucune reine ne puisse menacer une autre reine, c'est-à-dire qu'aucune paire de reines ne peut être placée sur la même ligne, la même colonne ou la même diagonale.



Ce problème est souvent utilisé comme exemple pour illustrer les concepts d'algorithme de recherche et d'optimisation. En effet, il existe plusieurs approches pour résoudre ce problème, chacune ayant ses avantages et ses inconvénients.

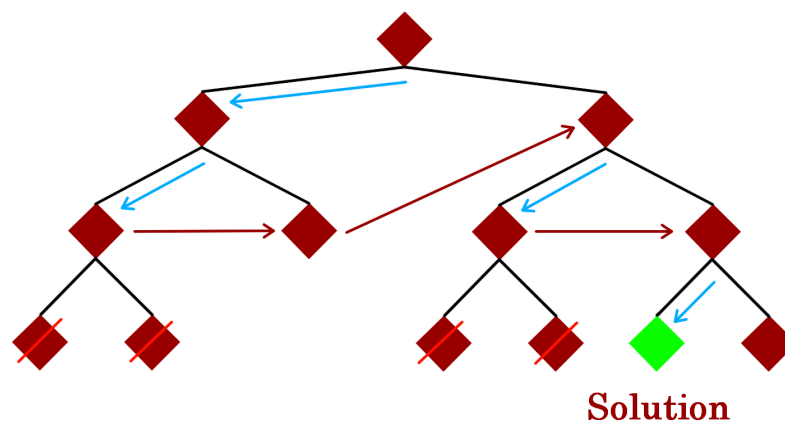
Dans ce compte rendu, nous allons examiner les différentes approches pour résoudre le problème des n reines, en nous concentrant sur les algorithmes de backtracking, force brute et de recherche locale. Nous allons comparer ces algorithmes en termes d'efficacité et de précision, et discuter des améliorations possibles pour résoudre ce problème.

Algorithme utilisé

L'algorithme de backtracking, également connu sous le nom de méthode de retour sur trace, est une méthode récursive de recherche de toutes les solutions d'un problème en explorant toutes les combinaisons possibles. Cet algorithme est très utilisé pour résoudre des problèmes combinatoires tels que le problème des n reines.

L'algorithme de backtracking commence par placer une reine dans la première colonne de la première rangée. A partir de la deuxième reine, il vérifie ensuite si cette reine peut menacer une autre reine déjà placée. Si c'est le cas, l'algorithme continue à explorer toutes les autres possibilités jusqu'à ce qu'il trouve une position valide pour la deuxième reine. Il place ensuite une reine dans la deuxième colonne de la deuxième rangée et continue à explorer toutes les possibilités jusqu'à ce qu'il trouve une position valide pour la deuxième reine, et ainsi de suite jusqu'à ce que toutes les n reines soient placées sur l'échiquier.

Si à tout moment l'algorithme rencontre une configuration qui ne peut pas être étendue à une solution valide, il revient en arrière et essaie une autre option. C'est pourquoi cette méthode est appelée méthode de retour sur trace.



Exemple d'exécution d'un algorithme de retour sur trace

L'algorithme de backtracking peut être implémenté de manière récursive ou itérative. L'implémentation récursive est plus simple et plus facile à comprendre, mais elle peut être moins efficace pour des valeurs élevées de n en raison de la pile d'appels récursifs. L'implémentation itérative est plus complexe, mais elle peut être plus efficace pour des valeurs élevées de n .

L'algorithme de backtracking est souvent utilisé comme point de départ pour résoudre des problèmes combinatoires plus complexes en ajoutant des heuristiques ou des techniques d'optimisation pour améliorer les performances.

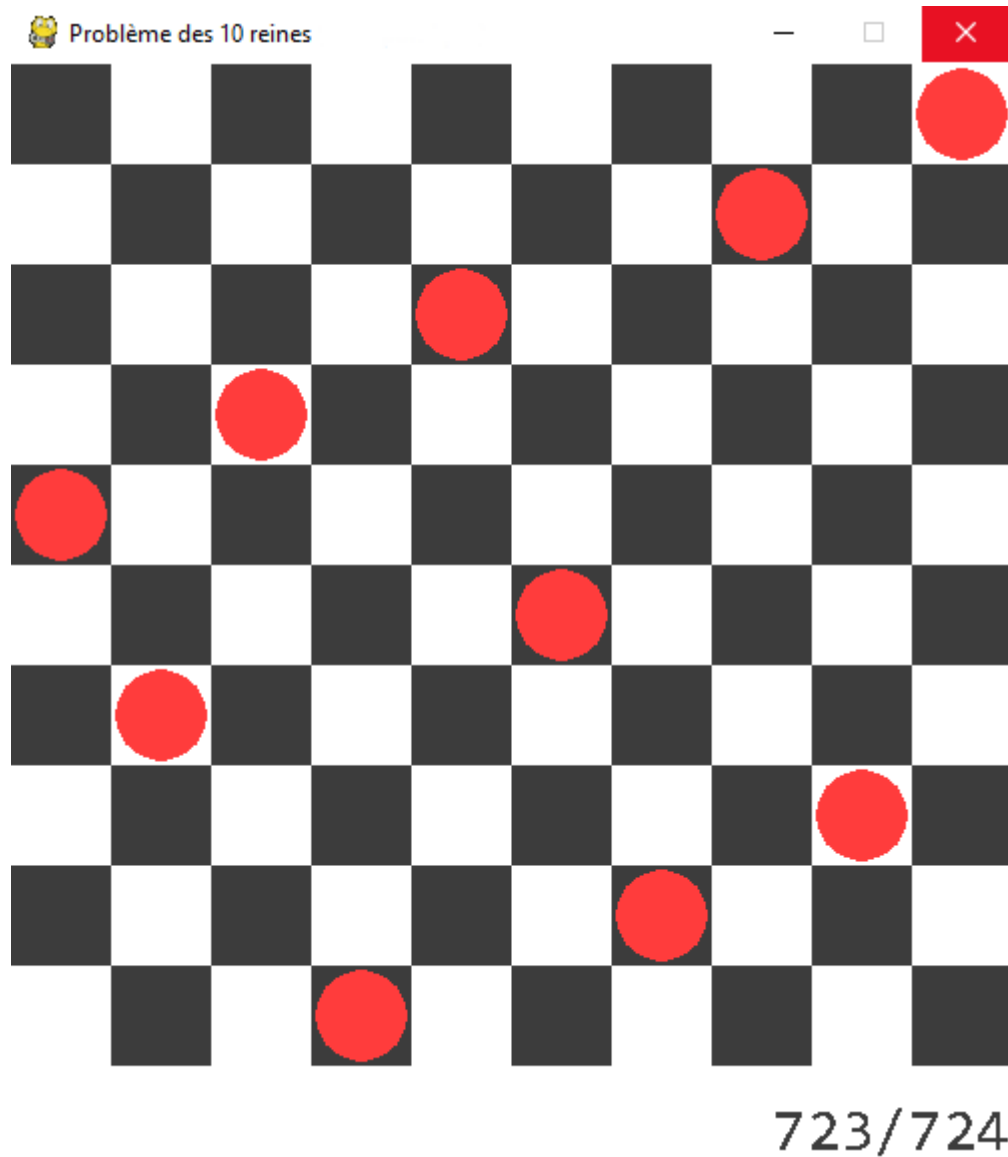
Programme python

Le programme python permet de choisir le nombre n de reines et calcule toutes les solutions grâce à l'algorithme de backtracking puis les affiche à la suite dans une fenêtre pygame.

Lien du Github : <https://github.com/HubertBDLB/SAE202>

Les instructions pour exécuter le programme sont sur le README.md

Résultat du programme pour $n = 10$:



Comparaison avec d'autres algorithmes

En comparaison avec l'algorithme de backtracking, l'algorithme de force brute peut également être utilisé pour résoudre le problème des n reines. L'algorithme de force brute teste toutes les combinaisons possibles de placement des reines sur l'échiquier et vérifie si chaque combinaison est valide. Bien que cette méthode garantit de trouver toutes les solutions possibles, elle peut être très inefficace pour des valeurs élevées de n , car le nombre de combinaisons possibles augmente exponentiellement avec n .

L'algorithme de recherche locale, en revanche, utilise une approche différente pour résoudre le problème des n reines. Au lieu d'explorer toutes les combinaisons possibles, il commence par une configuration initiale aléatoire et cherche à améliorer cette configuration en déplaçant les reines d'une position à une autre. Il continue à déplacer les reines jusqu'à ce qu'il ne puisse plus améliorer la configuration. Cette méthode peut être très efficace pour des valeurs élevées de n , car elle ne nécessite pas l'exploration de toutes les combinaisons possibles, mais elle ne garantit pas de trouver toutes les solutions possibles.

En comparaison avec l'algorithme de backtracking, l'algorithme de force brute est très inefficace pour des valeurs élevées de n , tandis que l'algorithme de recherche locale peut être très efficace mais ne garantit pas de trouver toutes les solutions possibles. L'algorithme de backtracking se situe quelque part entre les deux, car il explore toutes les combinaisons possibles, mais utilise des techniques de retour sur trace pour éviter d'explorer les branches inutiles. L'algorithme de backtracking est donc généralement plus efficace que l'algorithme de force brute pour des valeurs élevées de n , mais peut être moins efficace que l'algorithme de recherche locale pour certaines configurations d'entrée.

Conclusion

Le problème des n reines est un problème mathématique intéressant qui a des applications dans de nombreux domaines. Il existe plusieurs algorithmes pour résoudre ce problème, chacun ayant ses avantages et ses inconvénients.

L'algorithme de backtracking est le plus simple à comprendre et à implémenter, mais il est souvent très lent pour des valeurs élevées de n . L'algorithme génétique est plus efficace pour des valeurs élevées de n , mais il peut être plus difficile à mettre en œuvre et à optimiser. L'algorithme de recherche locale peut être plus rapide que l'algorithme de force brute, mais il peut être moins précis.

Des améliorations possibles pour résoudre le problème des n reines pourraient inclure l'utilisation d'algorithmes hybrides qui combinent plusieurs approches, ou l'utilisation de techniques d'optimisation plus avancées, telles que les algorithmes de colonie de fourmis ou les réseaux de neurones. Il est important de continuer à explorer de nouvelles approches pour résoudre ce problème et d'autres problèmes similaires.