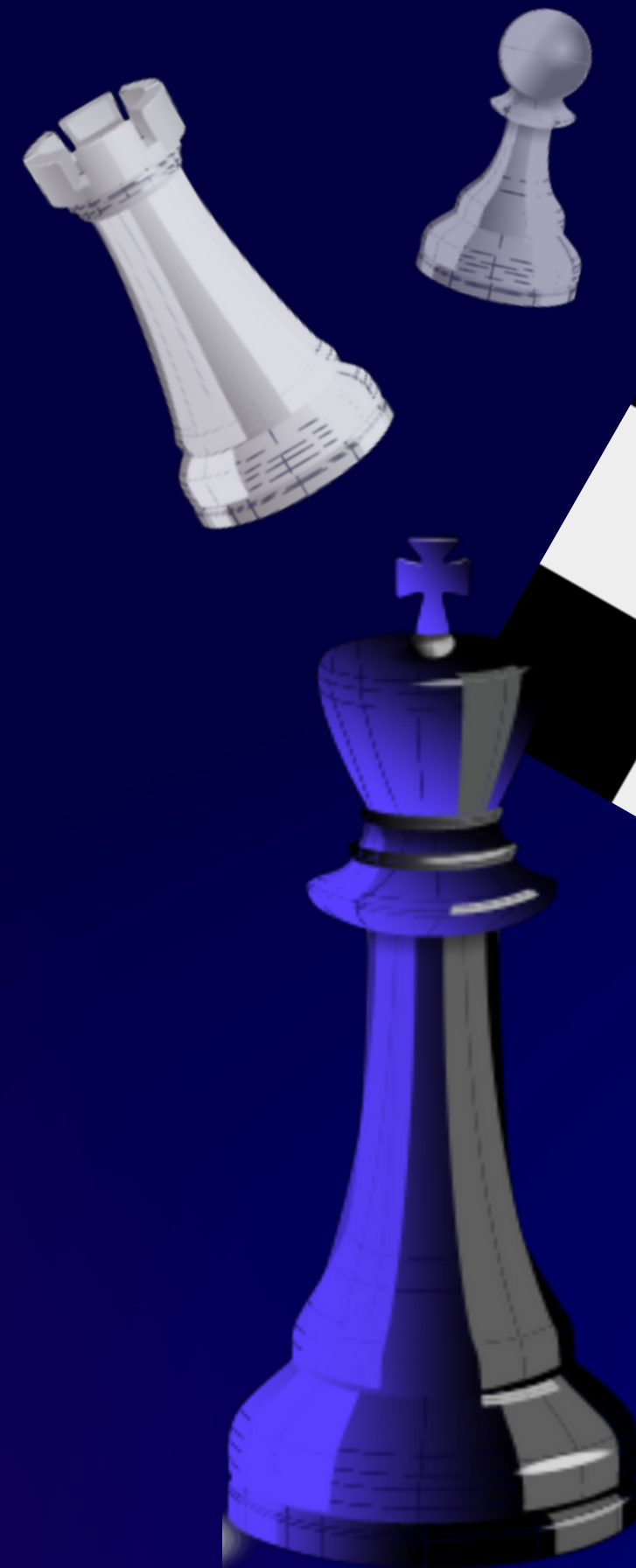


TP MÉTAHEURIQTIQUES

# PROBLÈME DES N-REINES

10/05/2023



**PRÉSENTÉ PAR**  
OUCHAR Manel  
YOUSFI Zakaria



# Sommaire



Introduction au  
problème



Méthodes de résolution



Algorithme génétique



Algorithme PSO



Résultats  
d'expérimentations

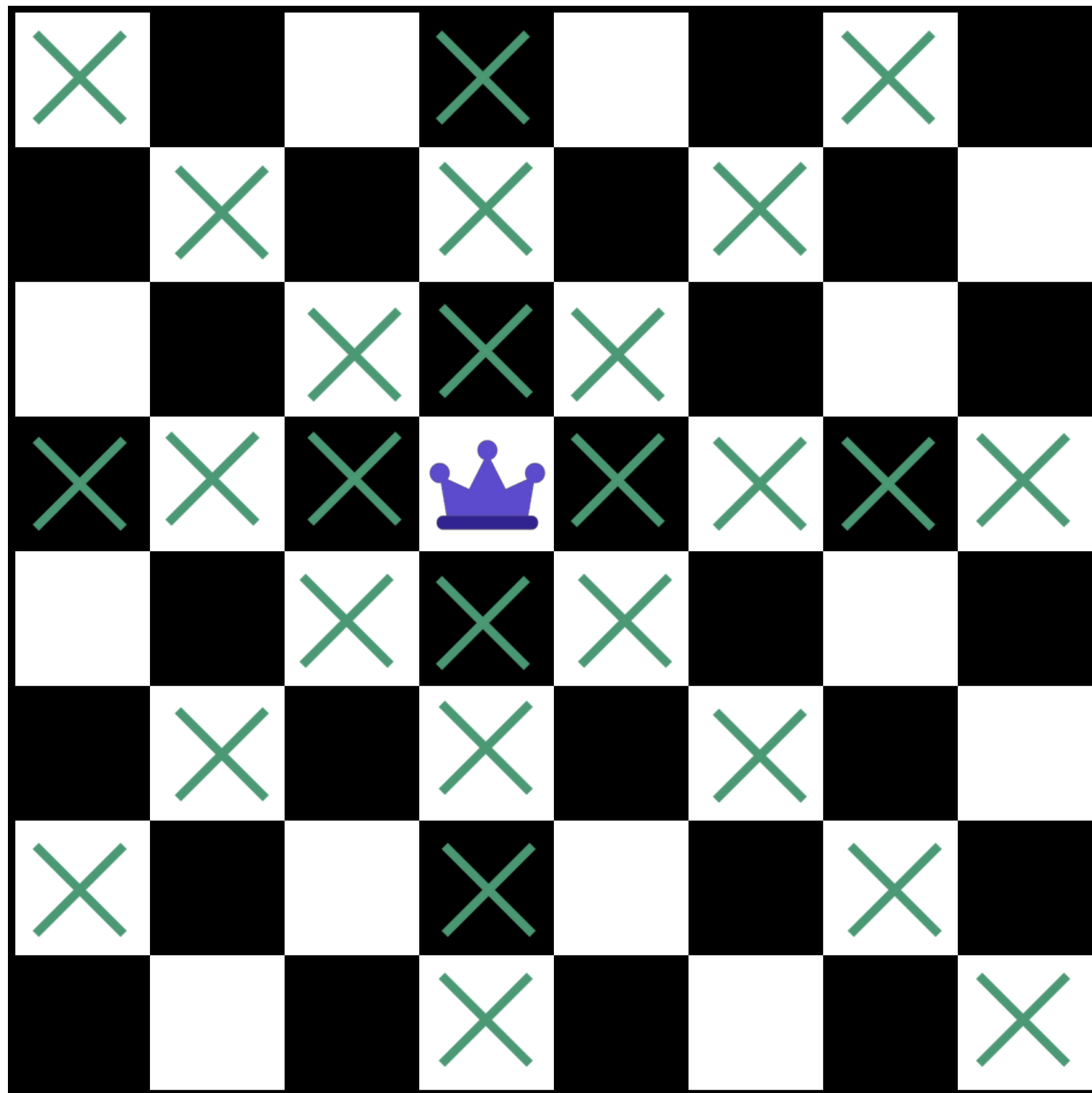


Etude comparative



Conclusion

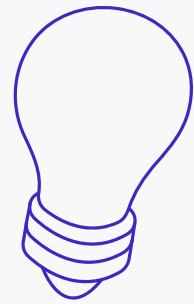




# Description du problème

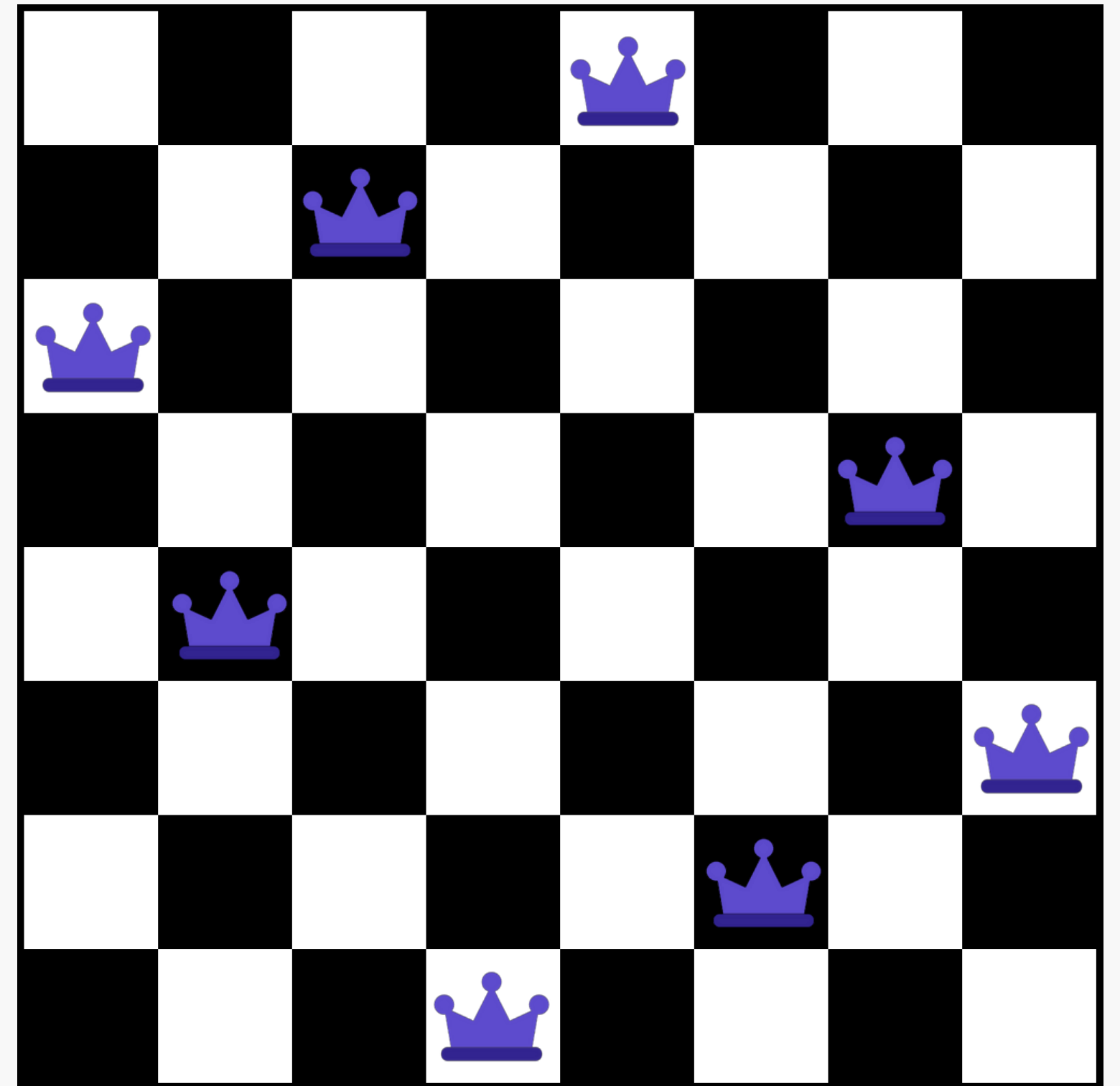
Étant donné un échiquier de taille  $N \times N$ , peut-on placer  $N$  reines sur l'échiquier de manière à ce qu'aucune reine ne menace une autre ?

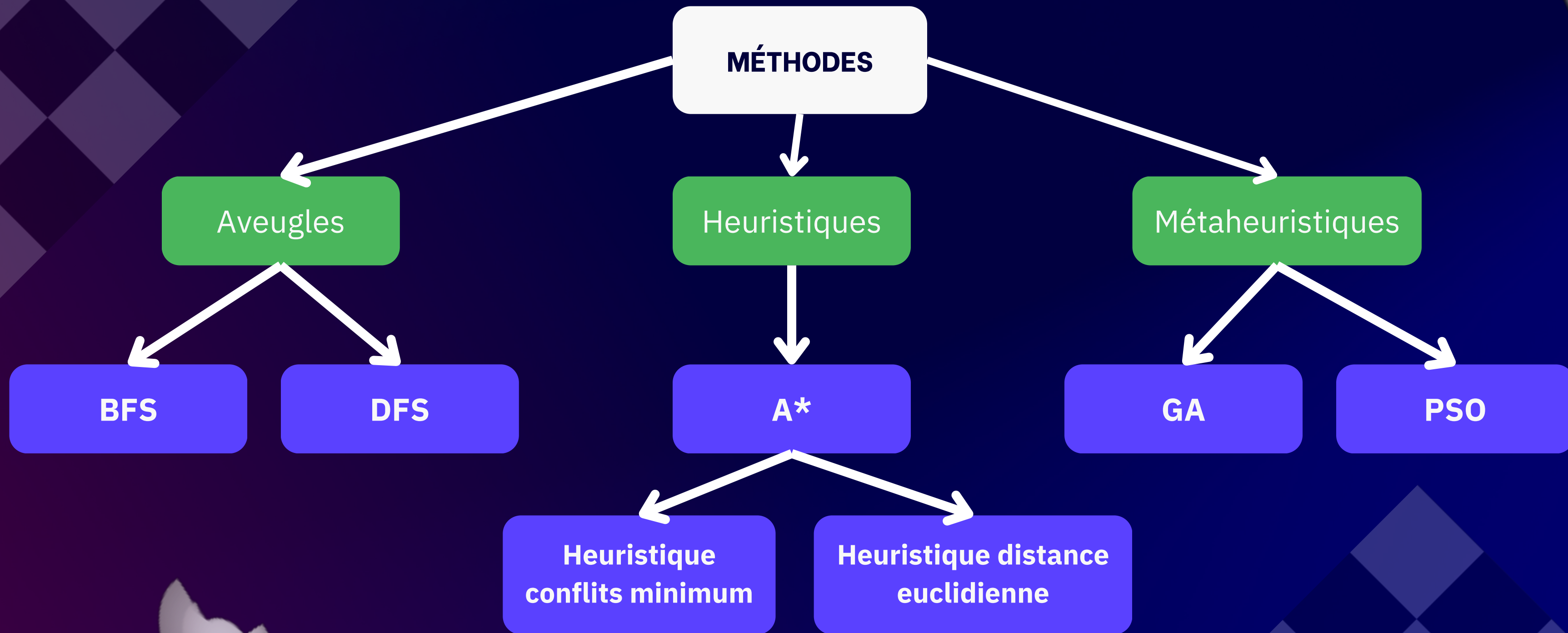
# OUI ! Mais...



Ce problème présente plusieurs défis :

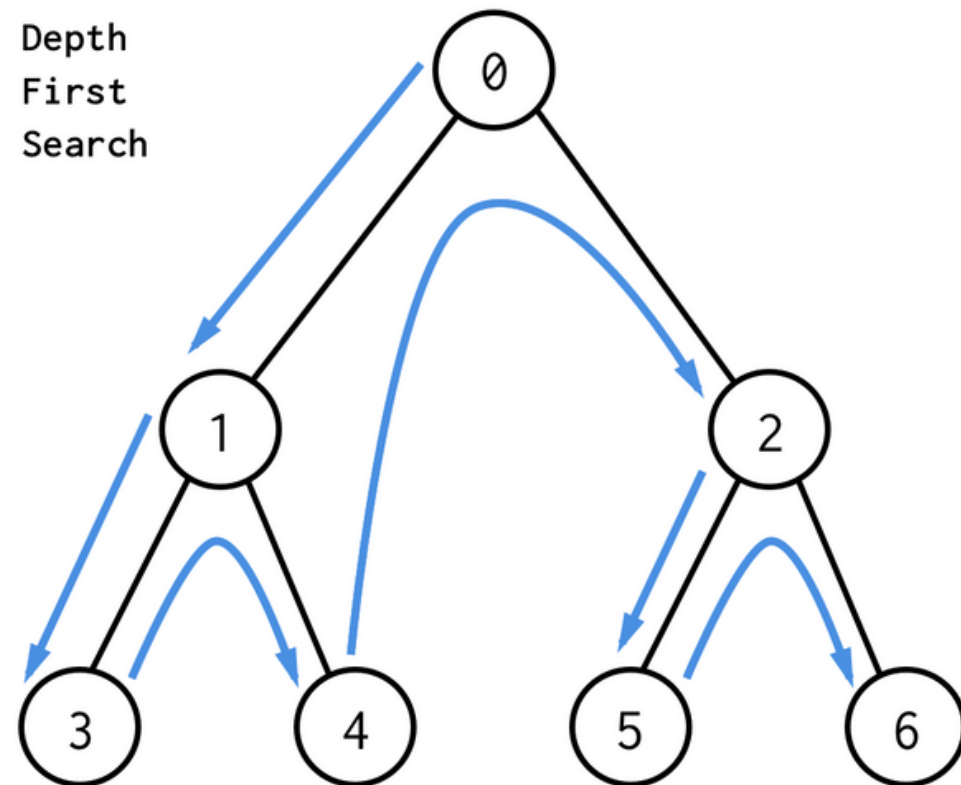
1. Complexité
2. Recherche de solution valide
3. Scalabilité



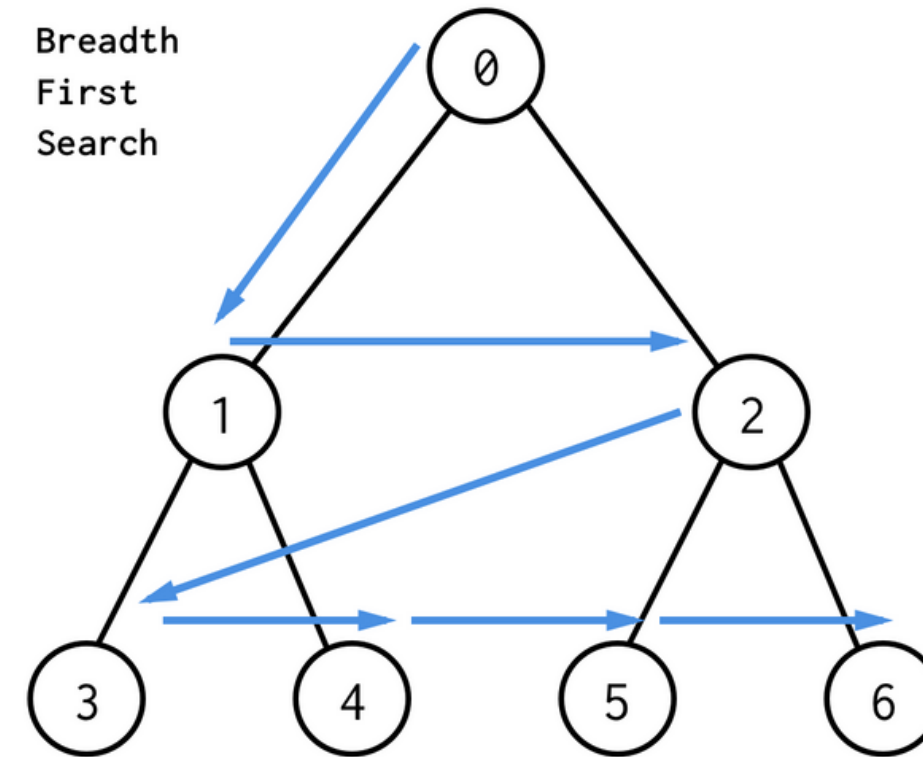




# DFS & BFS



DFS (Depth-First Search) est un algorithme de recherche qui explore un graphe en se déplaçant le plus profondément possible le long d'un chemin avant de revenir en arrière.

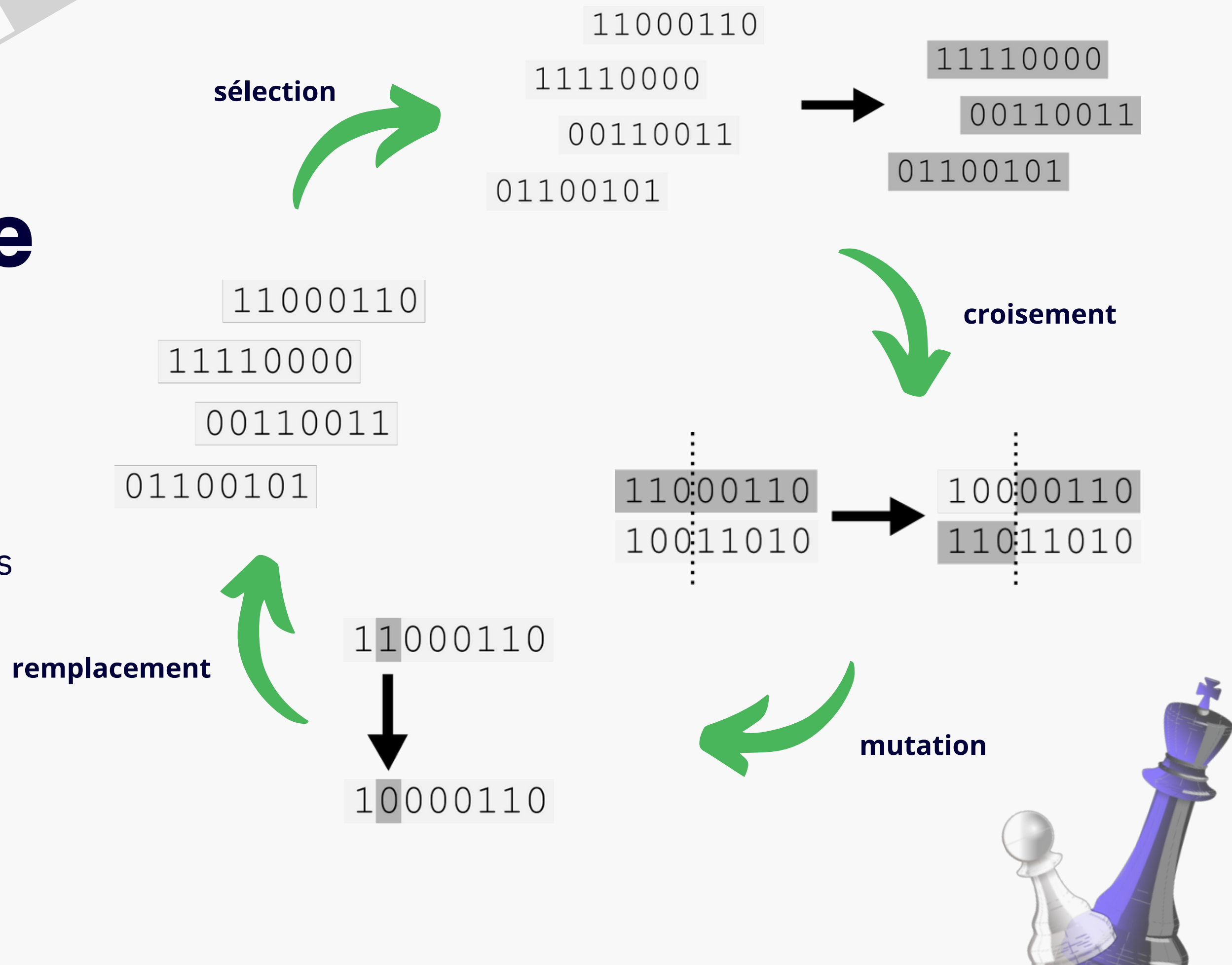


BFS (Breadth-First Search) est un algorithme de recherche qui explore un graphe en examinant tous les voisins d'un nœud avant de passer aux nœuds voisins suivants.



# L'Algorithme Génétique

Utilisation d'une population de solutions potentielles (appelées chromosomes), qui subissent des opérations de reproduction, de mutation et de sélection pour évoluer vers de meilleures solutions au fil du temps.

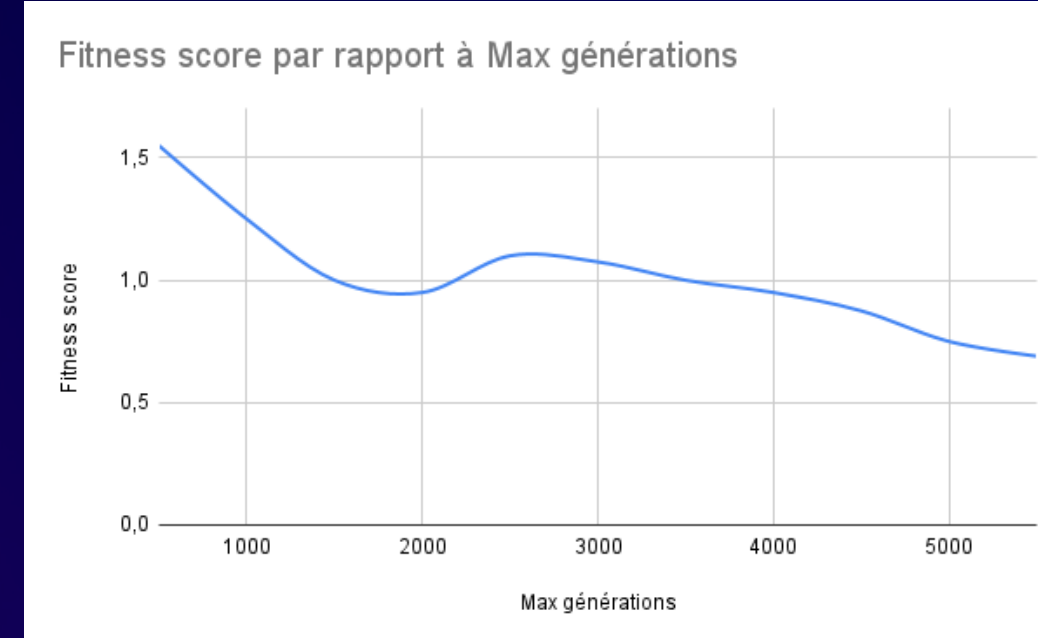
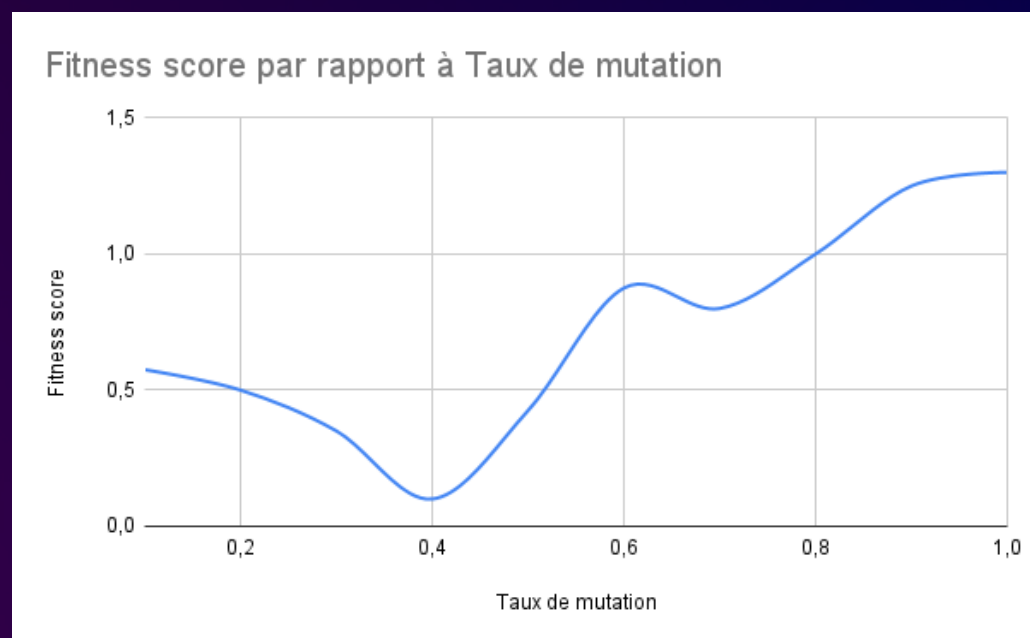
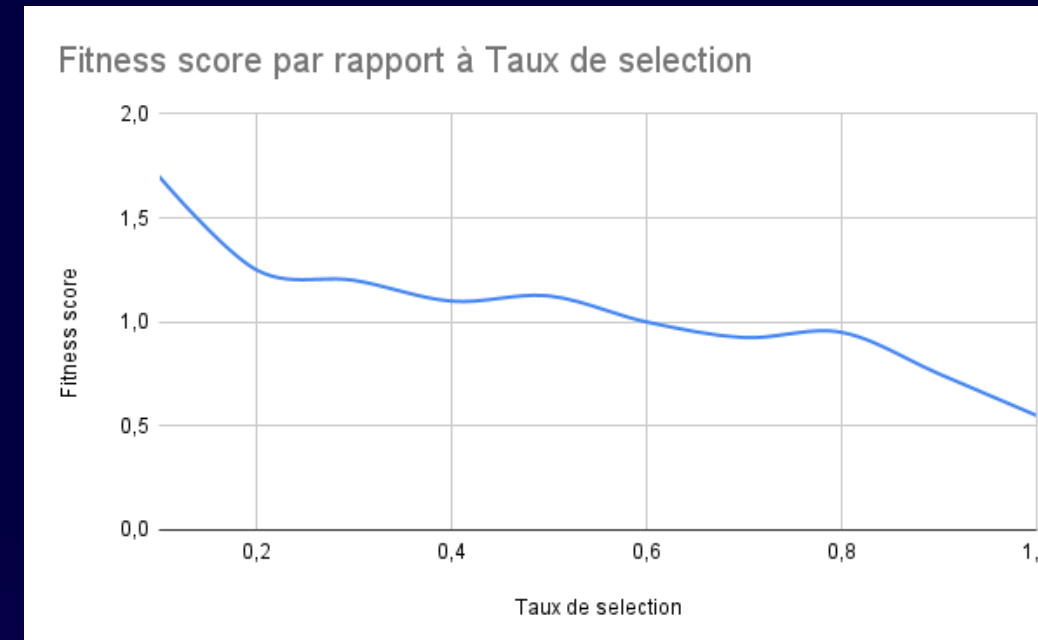
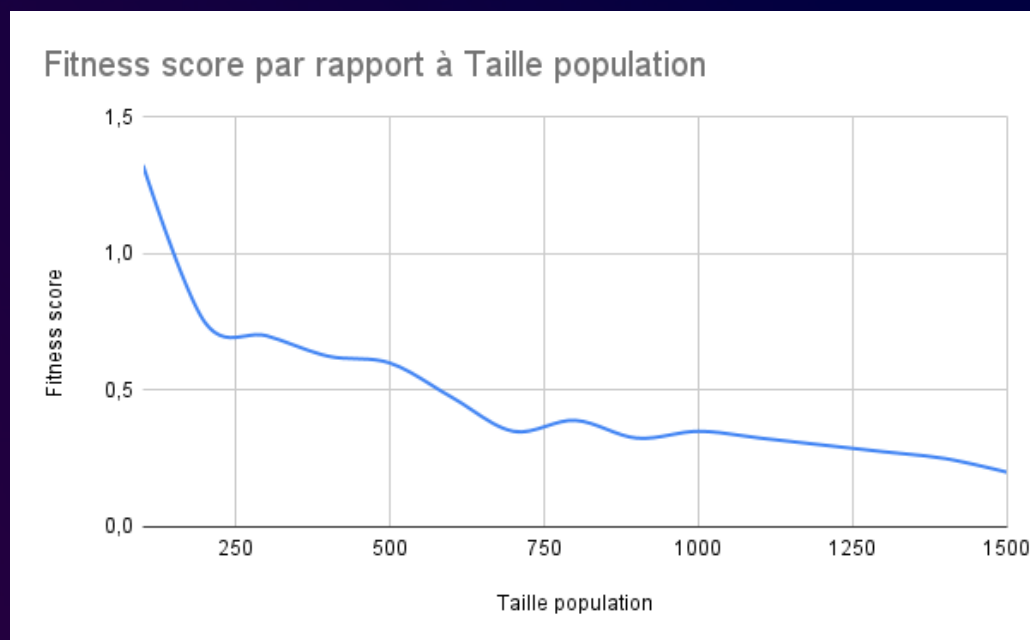


# Résultats d'expérimentations

## 1. Expérimentations sur les paramètres des métaheuristiques:



GA





# L'Algorithme Génétique

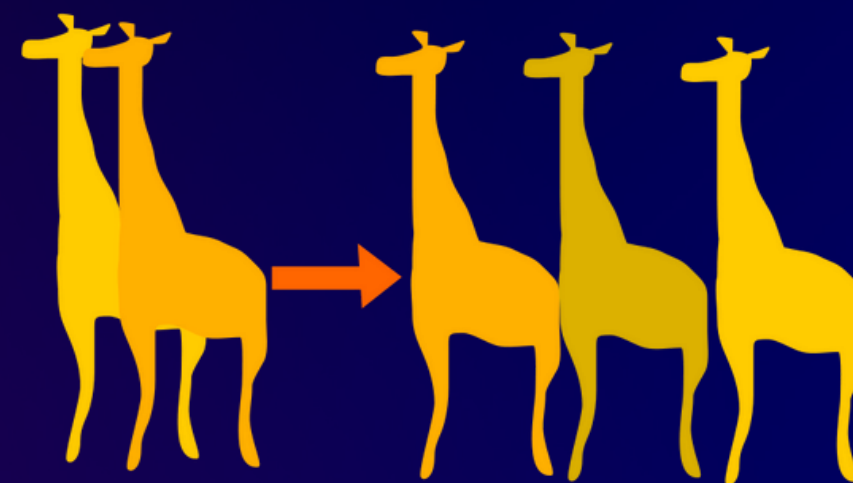
Les algorithmes génétiques sont inspirés de la théorie de Darwin qui repose sur trois principes : le principe de variation, le principe d'adaptation et le principe d'hérédité.



**principe de variation**



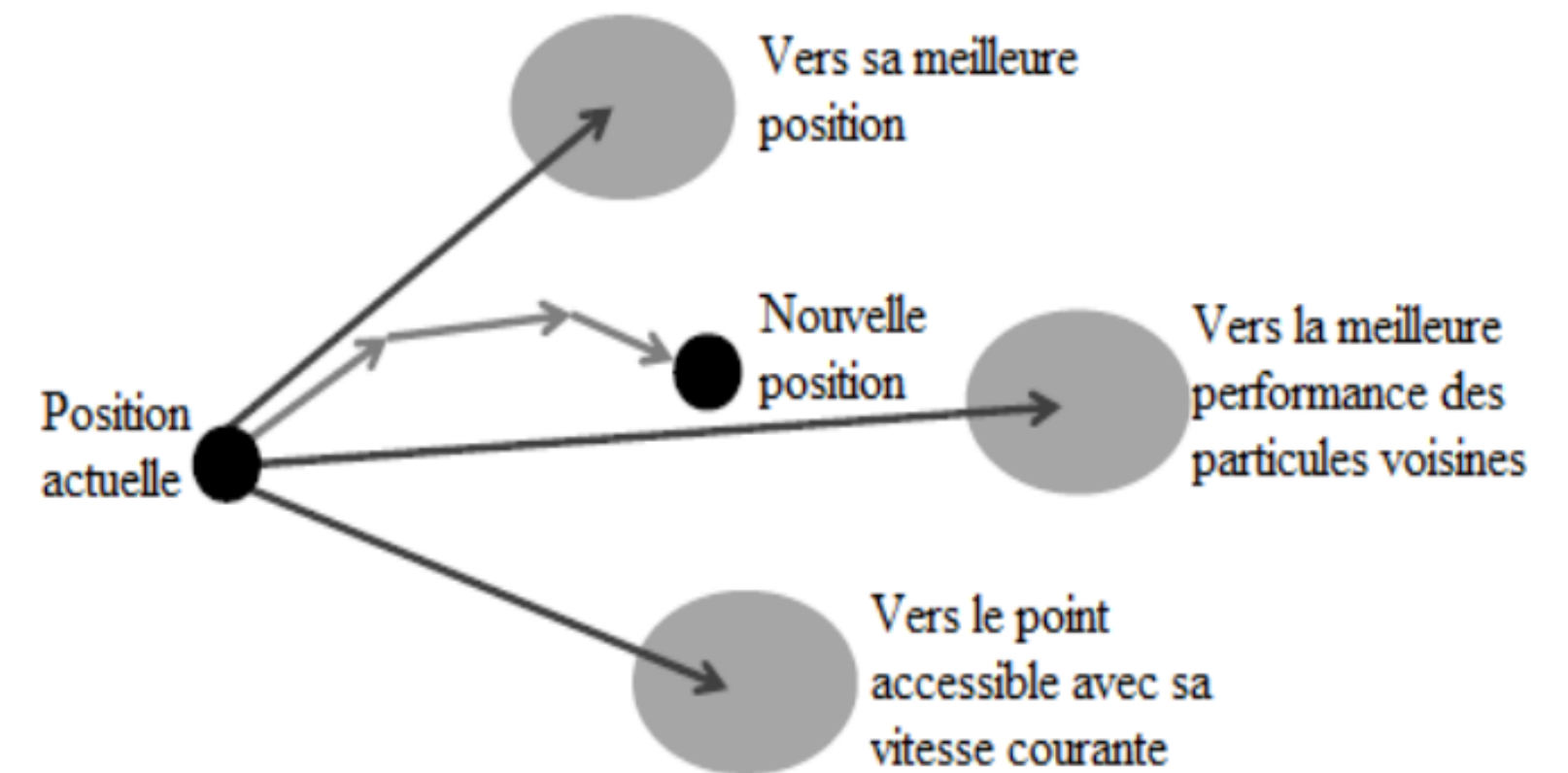
**principe d'adaptation**



**principe d'hérédité**



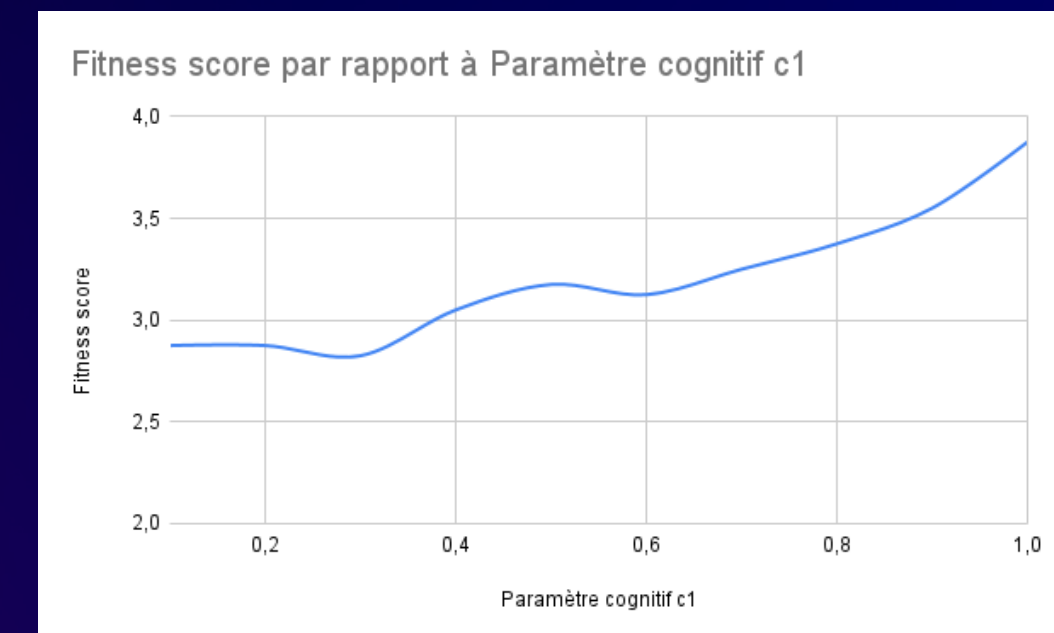
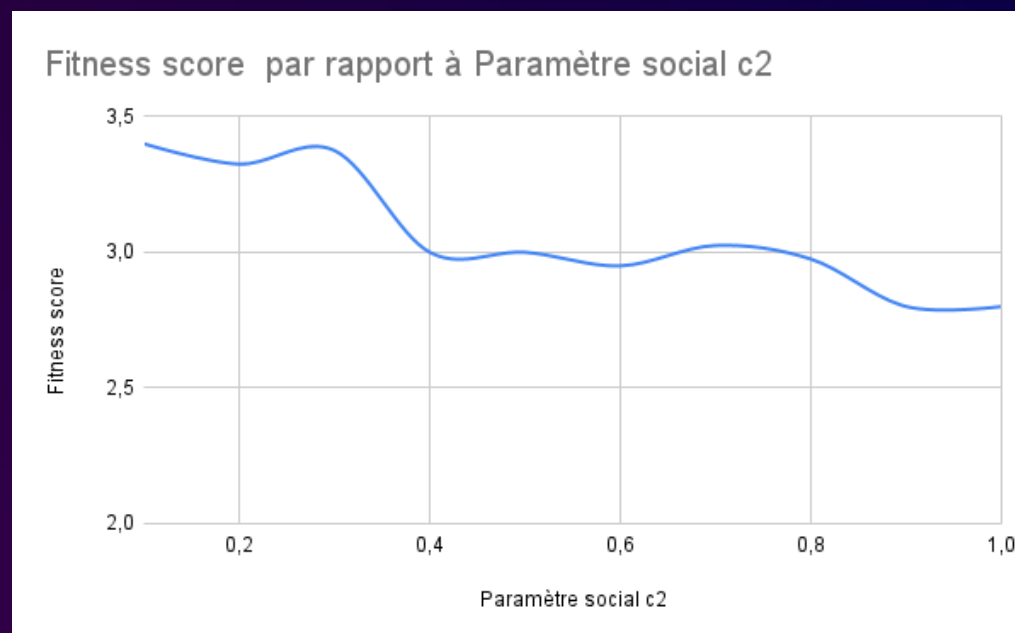
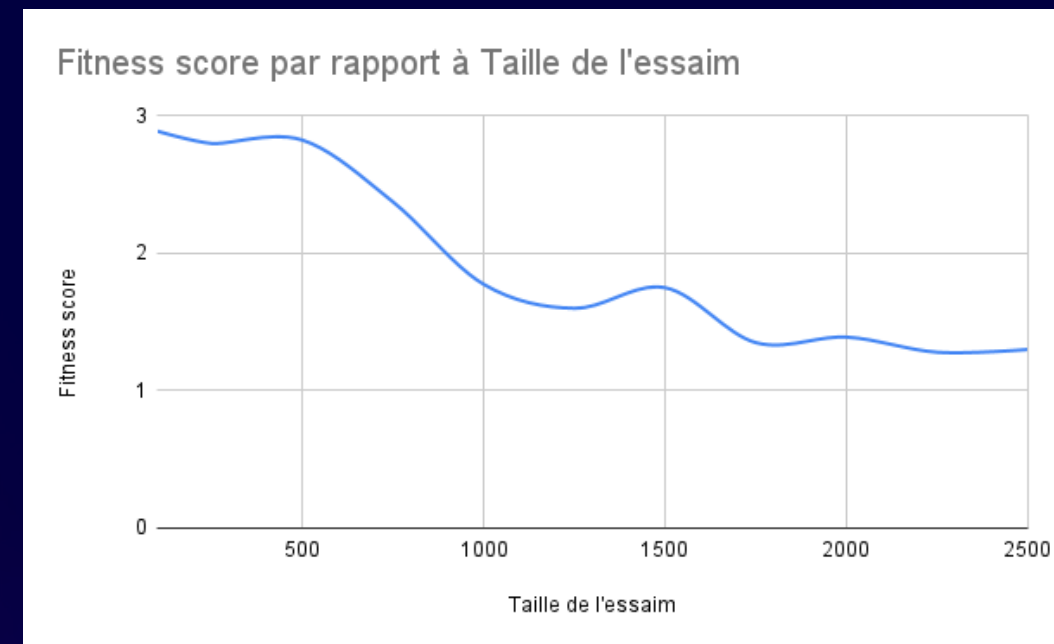
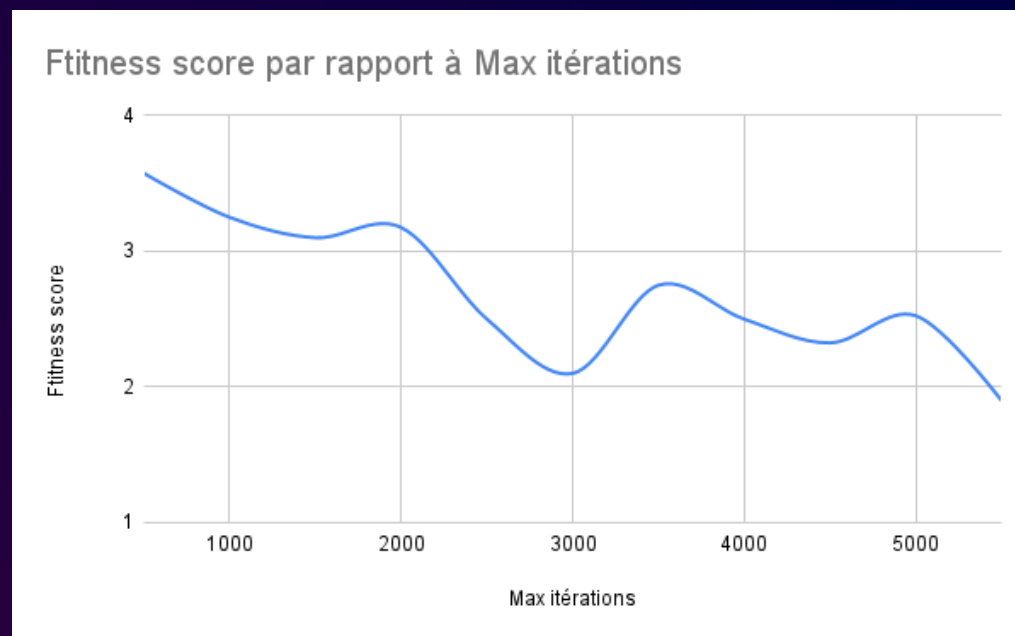
# L'Algorithme PSO



# Résultats d'expérimentations

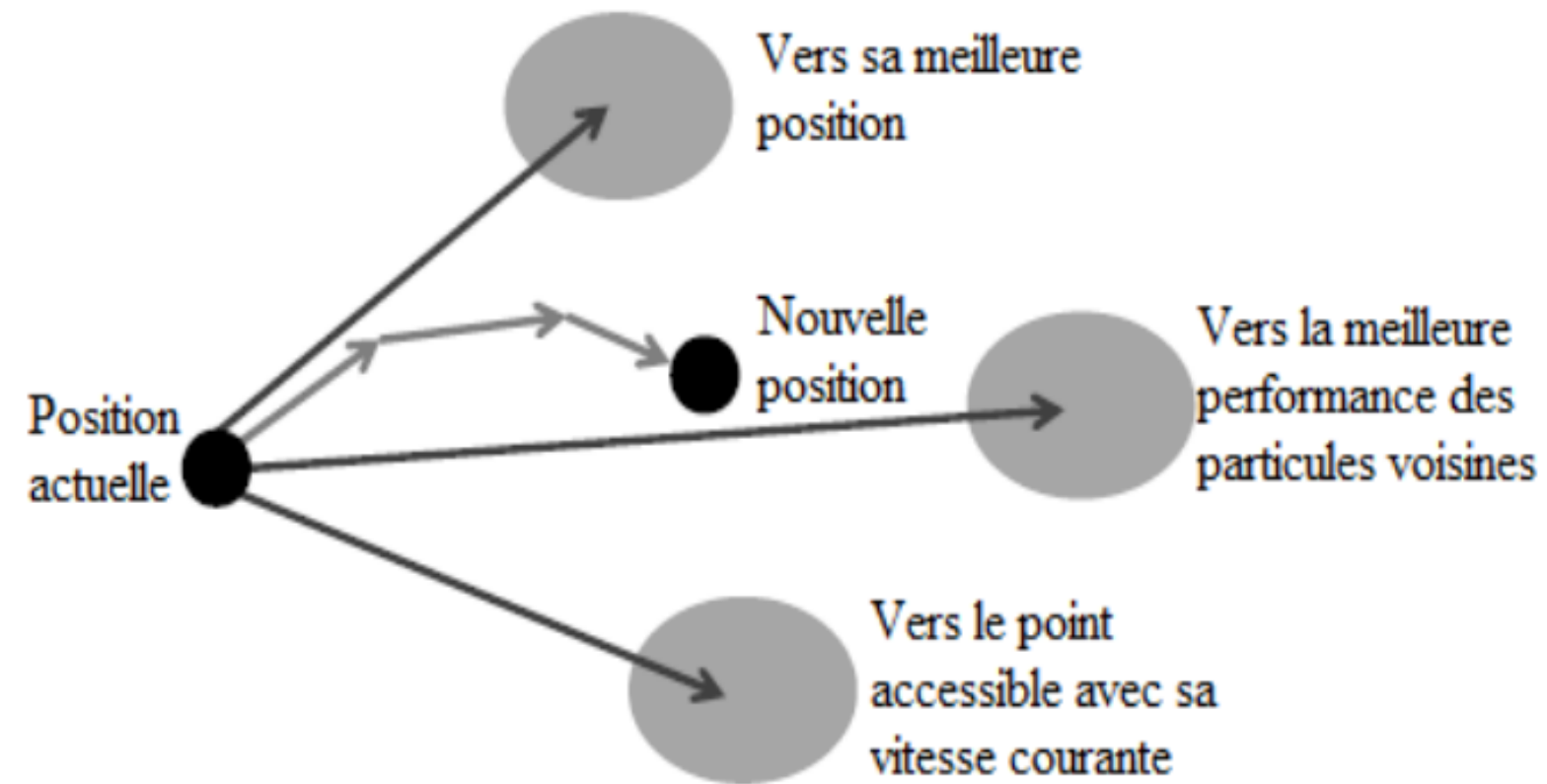
## 1. Expérimentations sur les paramètres des métaheuristiques:

PSO



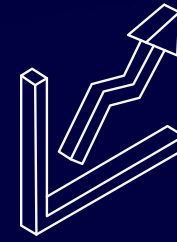
# L'Algorithme PSO

Les particules se déplacent dans un espace de solutions en ajustant leurs vitesses et leur position en fonction de leur meilleure performance individuelle et de la meilleure performance globale du groupe, favorisant ainsi la découverte de solutions optimales.





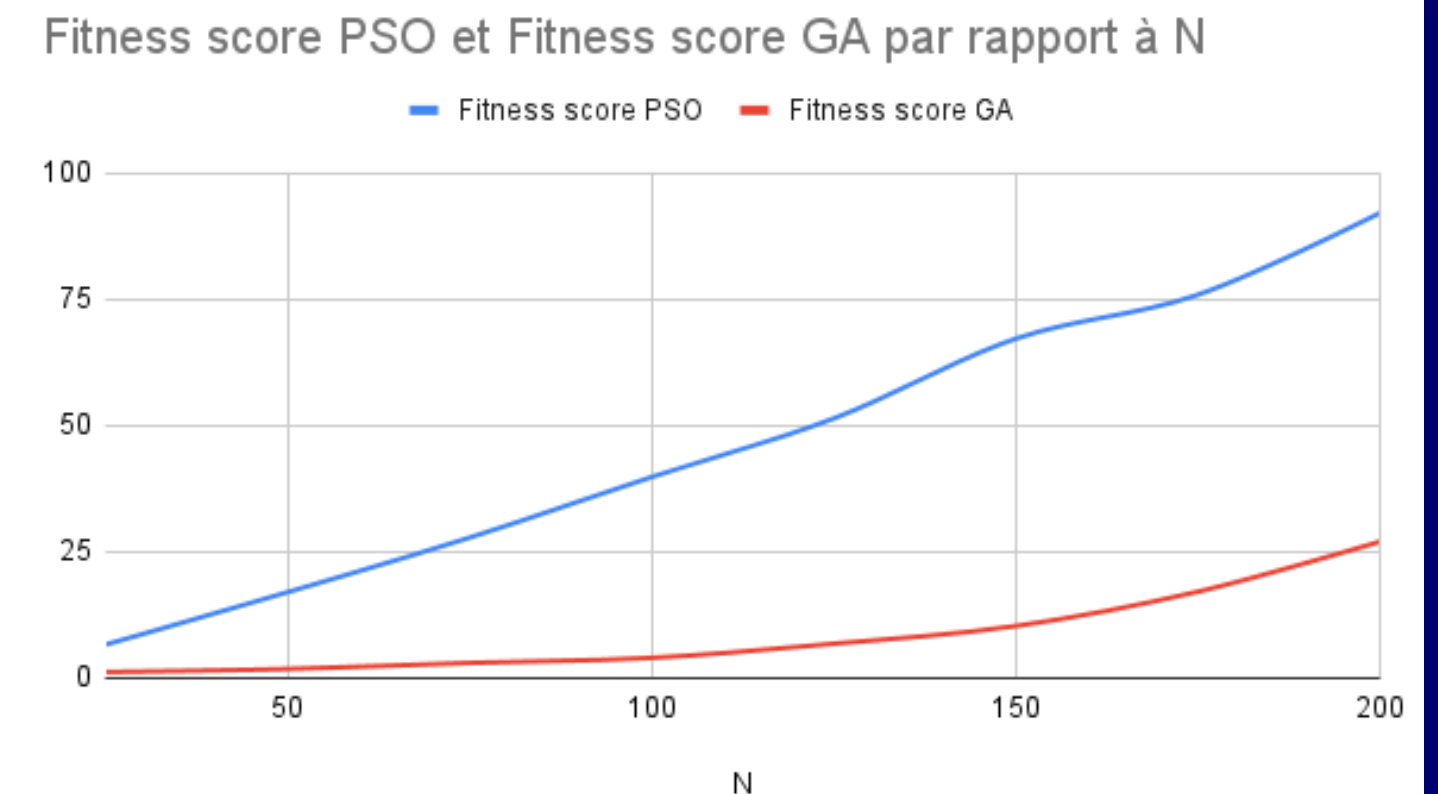
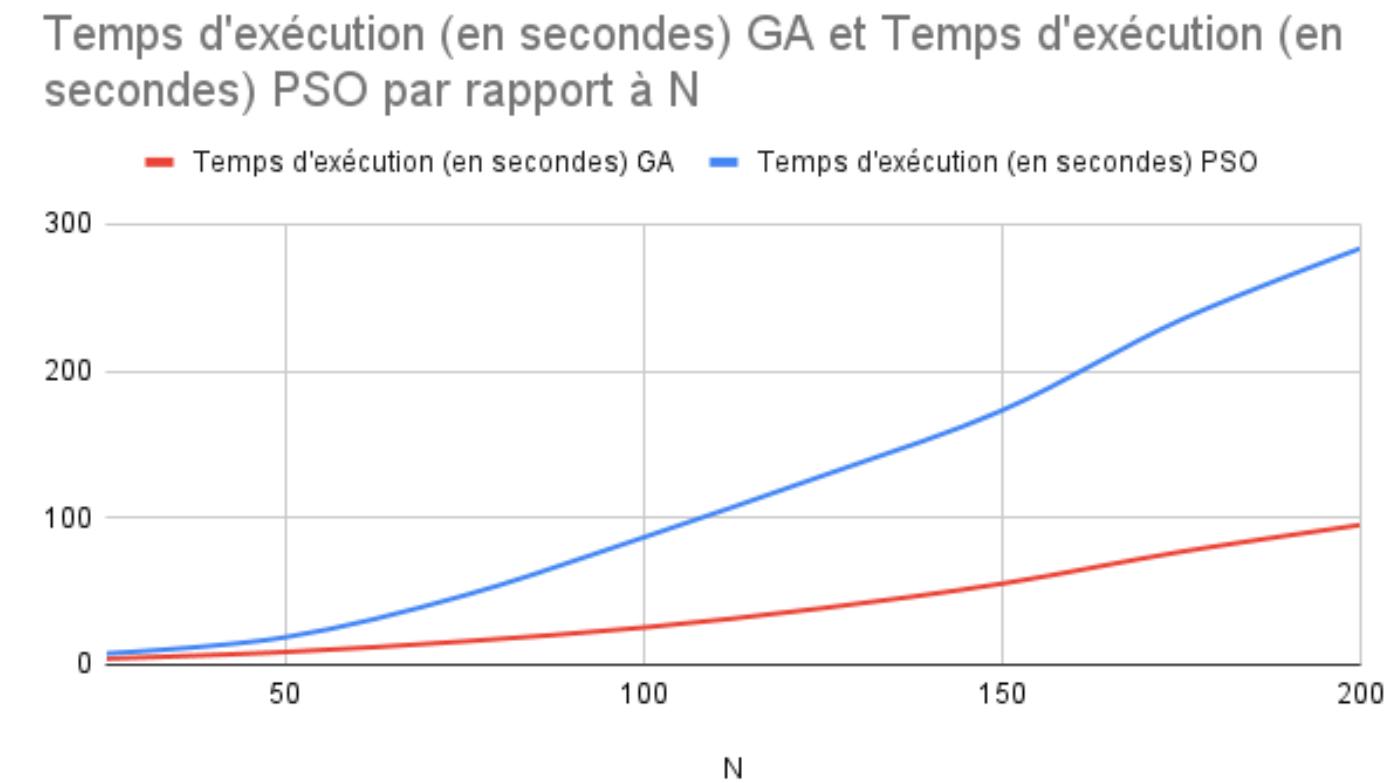
# Résultats d'expérimentations



## 2. Comparaison entre le performances du GA et du PSO:

La raison de l'augmentation du temps d'exécution et du fitness score est que plus la taille du problème augmente, plus l'espace de solutions augmente de manière exponentielle, ce qui rend plus difficile de trouver une bonne solution, et cela pour les deux algorithmes GA et PSO.

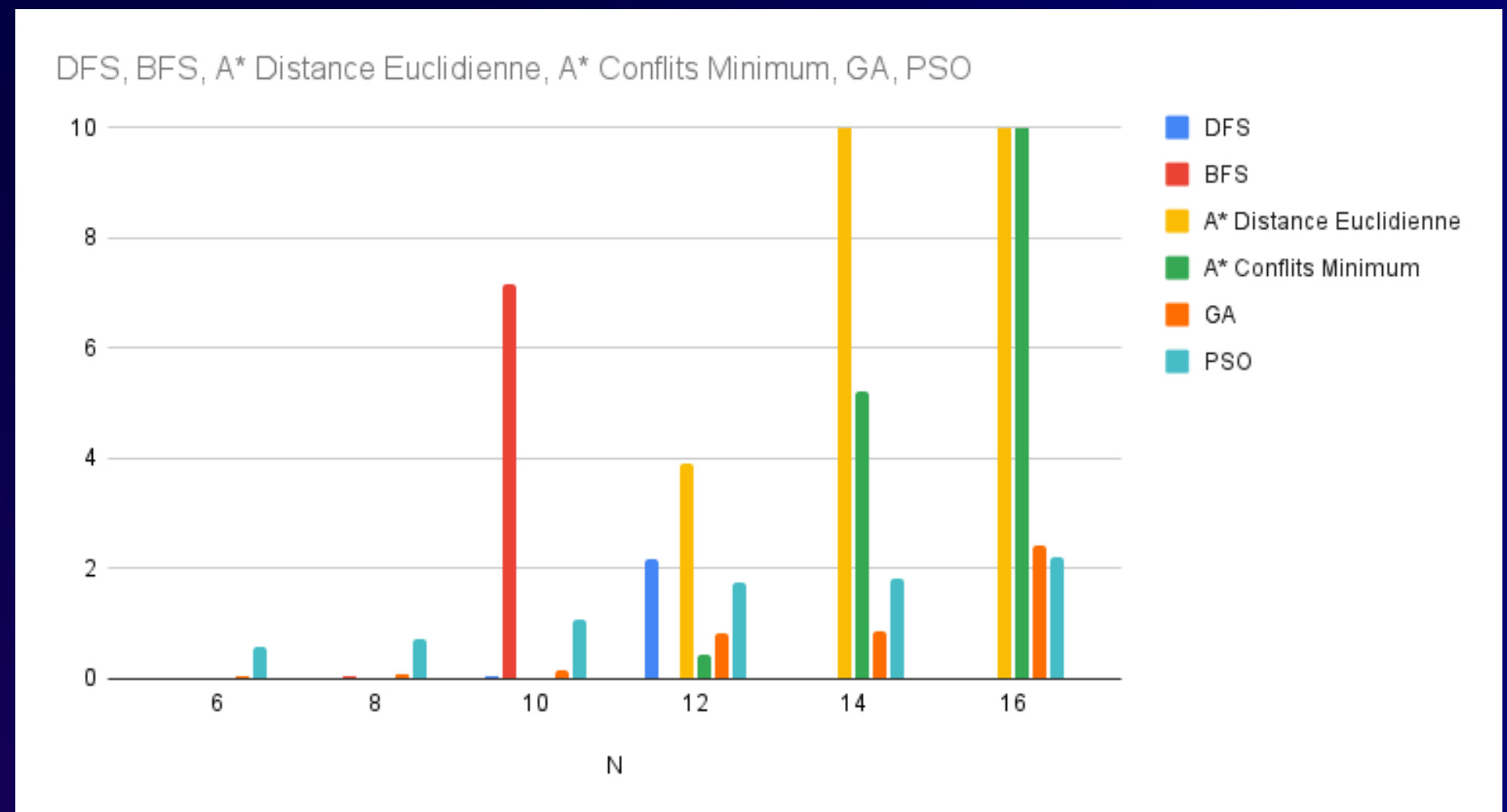
On remarque aussi que le GA surpasse le PSO pour les plus grandes tailles du problème.





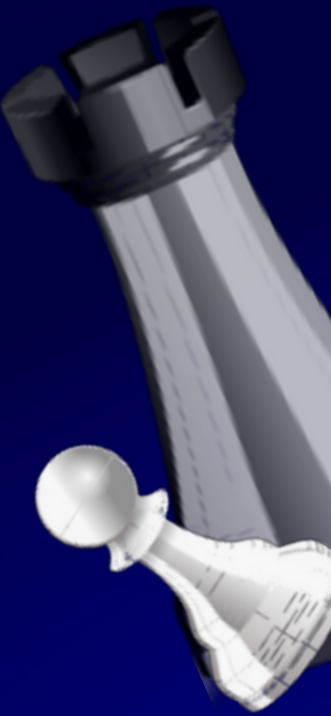
# Etude comparative

En comparant les différentes approches, on remarque que DFS et BFS sont rapides pour de petites valeurs de  $n$ , mais deviennent lents pour des valeurs plus grandes. A\* (Distance Euclidienne et Conflits Minimum) est meilleur que DFS et BFS pour des valeurs plus élevées de  $n$ , mais reste lent pour  $n > 14$ . Le GA est plus rapide et donne des résultats plus optimaux que le PSO, probablement en raison de son approche de recherche globale.



# Conclusion

Le choix de l'algorithme pour le problème des  $n$  reines dépend des exigences et des contraintes du problème. Les algorithmes de recherche basés sur l'état (DFS, BFS, A\*) sont complets mais peuvent être limités par la mémoire et l'espace de recherche pour de grandes tailles de problème. Les algorithmes métaheuristiques (GA, PSO) peuvent être plus efficaces pour de grandes tailles de problème, malgré l'absence de garantie d'optimum global. Le choix d'algorithme dépend des exigences spécifiques du problème et des ressources disponibles.



**Des questions ?**

