Intelligence Artificielle : TD/TP (Recherche locale)

Contexte :

Le but de ce TP est de programmer des méthodes de recherche locale pour résoudre le problème des N-Reines. Ce problème consiste à placer n reines sur un échiquier de taille (n x n) sans qu'elles s'attaquent : deux reines ne doivent pas se trouver sur la même ligne, sur la même colonne ou sur la même diagonale.

La figure 1 illustre une solution du problème avec 8 reines.

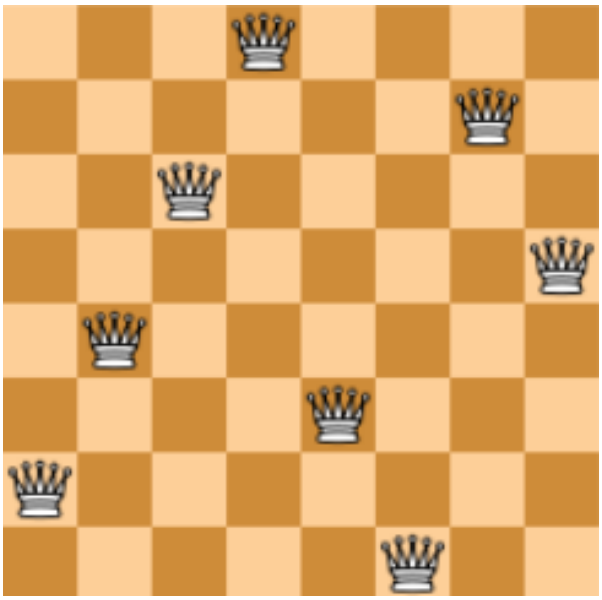

FIGURE 1 – solution du problème des 8-reines.

Implémentation

Un code java est disponible sur le site. Il contient 3 classes :

— **Node** : représente un noeud du problème (un état).
— **LocalSearchSolver** : c'est une classe qui implémente les algorithmes de recherche locales capables de résoudre le problème de N reines. Dans ce TP, on s'intéresse à deux algorithmes : *Hill Climbing* (ou méthode d'escalade) et *Simulated Annealing* (ou recuit simulé).
— **Main** : permet de tester les différents algorithmes implémentés.

Travail demandé

1. Complétez les méthodes suivantes de la classe **Node** :
   — *calculateCost()* : permet de calculer le coût du noeud (sa fonction objectif). Il s'agit du nombre de paires de reines qui s'attaquent.
   — *getBestNeighbor()* : retourne le meilleur voisin (successeur) immédiat du noeud.
   — *getRandomNeighbor()* : retourne un voisin immédiat aléatoire.
2. Implémentez la méthode *hillClimbing()* de la classe **LocalSearchSolver**.
3. Implémentez la méthode *simulatedAnnealing()* de la classe **LocalSearchSolver**.
4. Testez et analysez vos algorithmes dans le main.