UNIVERSITY OF SCIENCE

INFORMATION TECHNOLOGY BRANCH

LAB 01:

N-QUEENS PROBLEM

Course: CSC14003 - Artificial Intelligence

Student: Trịnh Hoàng An

Student’s ID: 21127577

HCMC – 07/11/2023

1. Table with statistics (Running time & Memory)

	Running time (s)			Memory (MB)		
Algorithms	N = 8	N = 100	N = 500	N = 8	N = 100	N = 500
UCS	Intractable	Intractable	Intractable	Intractable	Intractable	Intractable
A*	0.006948551	1828.117733	Intractable	91.16015625	1573.3203125	Intractable
GA	0.114650249	14.00202227	5241.270575	90.6875	91.19921875	106.7825521

2. Checklist:

Tasks	Done	Note
UCS Algorithms	x	
A* Algorithms	x	
Genetic Algorithms	x	
Simple GUI	x	
Data testing table	x	Some cases are “Intractable” where there is a large number of queeens
Brief description of main functions	x	I have note in my code
Report	x	

3. Meaningful observations and comments on the statistics

3.1 About Uniform-cost search

Running time and memory usage information for UCS is "intractable" for all N values (8, 100, 500), we can conclude that UCS is not suitable for N-Queen problem with large size. Or it's because your algorithm is not fully optimized, but through that we can conclude that with the requirements of the problem, the UCS algorithm is very infeasible.About A* with MIN-CONFLICT heuristic

3.2 About A* with MIN-CONFLICT heuristic

- With N=8: Algorithm A* with heuristics MIN-CONFLICT can solve N-Queen problem with 8x8 chess board. The MIN-CONFLICT heuristics uses a conflict resolution strategy to search for optimal approximations. However, in this case, A* can run into overflow problems if not done carefully, since the number of states grows rapidly

- With N=100: A* with heuristics MIN-CONFLICT can apply and solve the N-Queen problem with a 100x100 chessboard. The MIN-CONFLICT heuristics greatly reduce the number of states traversed and increase the performance of the algorithm.
- With N=500: A* with heuristics MIN-CONFLICT cannot solve the N-Queen problem with a 500x500 chessboard. With such a large number of states, calculating and traversing the states becomes infeasible in reasonable time and consumes a lot of memory.

⇨ The running time of A* increases significantly as the value of N increases. With N=8, the runtime is only 0.006948551 seconds, while with N=100, the runtime is 1828.117733 seconds (about 30 minutes). This shows that A* is not efficient when applied directly to N-Queen problems with large sizes such as N=100 and 500. In particular, for N=500, the running time of A* is considered "intractable" ", i.e. not feasible.

3.3 Genetic algorithm with the objective function defined from the above heuristic.

With N = 8:

- GA runtime is 0.114650249 seconds.
- Memory usage of GA is 90.6875 MB.
- With N=8, GA can solve the N-Queen problem quickly and efficiently. Runtime and memory usage are both acceptable.

With N = 100:

- GA runtime is 14.00202227 seconds (about 14 seconds).
- Memory usage of GA is 91.19921875 MB.
- With N=100, GA can still solve the N-Queen problem quite efficiently. Run time increased from N=8, but still acceptable. Memory usage increased slightly, but still at a fairly low level.

With N = 500:

- GA runtime is 5241.270575 seconds (about 1 hour and 27 minutes).
- Memory usage of GA is 106.7825521 MB.
- For N=500, GA can still solve the N-Queen problem, however, the running time has increased significantly and becomes very long. Memory usage also increased, but still pretty low compared to run time.

⇨ In general, GA is an algorithm capable of solving N-Queen problems of rather large size. With N=8 and 100, GA has good performance and has acceptable runtime and memory usage. However, with N=500, the running time of GA increases significantly and has a long processing time, although the memory usage is still quite low. Using GA for N-Queen problems with a size as large as N=500 requires patience and considerable computational resources.

4. Summary

Based on the requirements of the N-Queen problem, where we need to find a configuration such that N queens do not conflict and each queen is on a separate column, we can draw the following conclusions:

- UCS (Uniform Cost Search): Not suitable for N-Queen problems with large sizes like N=8, 100, 500. Computational complexity increases exponentially and UCS requires storing the entire state traversed, leading to a lot of memory and processing time.
- A* (A-star): A* can be used for N-Queen problems with N=8 and 100, but not suitable for N=500. A* with the MIN-CONFLICT heuristics can solve the problem with relatively good performance for small values of N, but running time and memory usage increase significantly as N increases.
- GA (Genetic Algorithm): Suitable for N-Queen problems with N=8, 100 and 500. GA is capable of solving problems with larger size and can find near-optimal solutions. However, running time and computational resources increase when N is large, requiring considerable patience and computational resources.

Summary :

- For the N-Queen problem with N=8, both A* and GA are suitable and can be used to find the solution.
- For the N-Queen problem with N=100, both A* and GA are capable of solving, but GA has better running time and computational resources.
- For the N-Queen problem with N=500, GA is the most suitable choice because A* becomes unfeasible in reasonable time and UCS requires storing too many states.

NOTE : However, in different cases, it is necessary to consider the time requirements, computational resources and performance of the algorithm to choose the most suitable method for the N-Queen problem.