Artificial Intelligence

Program – The n-Queen Problem

EECS 26' You-Ting Liu 111060013

Method

Board Representing

The board state is encoded as a list of integers, where the index represents the row and the value at that index represents the column. For example, [0, 1, 2, 3, 4] represents the following board:

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

Iterative Deepening Search (IDS)

The overall structure consists of a depth-first search, which is wrapped inside a for-loop that iterates over increasing maximum depth values. The initial state is an empty board, represented by a list of length board size, where each element is set to board size: [board size, board size, board size, …]. To formulate the problem, place a queen in each row one by one. A valid successor state is defined as placing a queen in the next row without causing any attacks between queens, while the depth remains less than the maximum depth limit. However, it is obvious that the solution must be found at a depth equal to the board size. Therefore, the iterative max depth limit in depth-first search is excessive, leading to wasted time.

Hill Climbing Algorithm (HC)

An initial state is generated by randomly shuffling a list of integers from 0 to board size - 1. A successor state is obtained by selecting any two elements and swapping them, provided that the number of attacks does not increase. To avoid getting stuck on a plateau, states with fewer attacks are prioritized. If multiple states have the same number of attacks, the one with the least is chosen. If no improvement is possible, a successor is randomly selected from states with an equal number of attacks. This randomness helps prevent the algorithm from getting stuck at a local optimum, as it allows for exploration of alternative solutions when no immediate improvement is available.

Genetic Algorithm (GA)

The fitness of a board is determined by the number of non-attacking queen pairs, which is calculated as the total possible attacking queen pairs minus the number of attacking pairs. The selection of parents is performed using tournament selection, where a subset of boards is randomly chosen, and the two with the highest fitness are selected as parents. The crossover operation creates two new child solutions by combining parts of two parent solutions at a randomly chosen point. The mutation operation introduces diversity by randomly swapping two positions or replacing an element with an unused value. This approach ensures that missing values (disappeared genes) are reintroduced into the solution, preventing duplicate numbers and preserving the integrity of the gene set. By avoiding loss of diversity, the algorithm can still converge towards an optimal solution rather than getting stuck in invalid or suboptimal states, helping to escape local optima and maintain genetic diversity within the population.

Comparison

Iterative Deepening Search (IDS)

Iterative deepening search offers stable efficiency since it systematically explores the search space, avoiding the randomness that can impact other algorithms. However, due to the large search space of the n-Queen problem, even with optimizations to reduce unnecessary exploration, iterative deepening search tends to be less efficient than algorithms that leverage randomness, such as genetic algorithm and hill climbing algorithm.

Hill Climbing Algorithm (HC)

Hill climbing algorithm demonstrates the best efficiency and the best success rate among the three approaches. By incorporating random choices in plateau regions, hill climbing algorithm effectively avoids getting stuck in local optima, reducing the number of required restarts. This adaptation significantly improves its performance, making it the most effective solution for the problem. Additionally, hill climbing algorithm is inherently simple, with a minimal amount of code required for implementation. This simplicity allows it to outperform genetic algorithm when dealing with smaller problem sizes, where the benefits of evolutionary processes are less pronounced. However, despite its efficiency, hill climbing algorithm still has the risk of being trapped in local optima if the random selection strategy is not properly tuned.

Genetic Algorithm (GA)

Genetic algorithm benefits from its ability to explore a diverse search space through an initial population that avoids row conflicts. This characteristic allows genetic algorithm to converge efficiently toward valid solutions. Since the first generation consists of 1000 individuals, the probability of already having a valid solution is high, leading to good execution efficiency. However, genetic algorithm requires careful tuning of various parameters, such as mutation rate and selection strategy, to ensure optimal performance. While it is a powerful approach for larger-scale problems, hill climbing algorithm often outperforms genetic algorithm in smaller problem sizes due to its more straightforward search strategy.

Conclusion

Overall, hill climbing algorithm performs the best in the n-Queen problem, while iterative deepening search is limited by the size of the search space, making it less efficient. Genetic algorithm improves solution search efficiency through its initial population design, but hill climbing algorithm may have an advantage when the problem size is smaller.

Test Results

8-Queen Problem

Table 1: The result of 8-queen problem

Algorithm	IDS	HC	GA
Time limit (s)	600	600	600
Trials	100	100	100
Average # attacks	0	0	0
Average running time (ms)	19.304	1.058	7.883
Success rate	100%	100%	100%

50-Queen Problem

Table 2: The result of 50-queen problem

Algorithm	IDS	HC	GA
Time limit (s)	600	600	600
Trials	1	30	30
Average # attacks	990	0	0.034
Average running time (s)	600.046	9.254	71.183
Success rate	0%	100%	96.67%

Attributes Settings

Iterative Deepening Search (IDS)

Table 3: The attributes settings of iterative deepening search

Attributes	Value
Initial depth limit	1
Depth limit increment after each DFS	1

Hill Climbing Algorithm (HC)

Table 4: The attributes settings of hill climbing algorithm

Attributes	Value
Restart times limit	1000000000000

Genetic Algorithm (GA)

Table 5: The attributes settings of genetic algorithm

Attributes	Value
GA type	Customized
Representation	Integer
Population size	1000
Selection	10-tournament
Crossover	1-point
Crossover rate	1
Mutation	Swap or insert missing value
Mutation rate	0.1
Survivor	Delete-oldest
Termination	Time limit 600s

Other Tests

DFS Is Better Than IDS in This Problem

Obviously, all solutions must be at depth N when there are N queens. Using pure DFS is more efficient than IDS because the depth limits from 1 to N-1 are unnecessary. Below are the results of these two algorithms for the 25-queen problem.

Table 6: The result of 25-queen problem

Algorithm	DFS	IDS
Time limit (s)	600	600
Trials	1	1
Average # attacks	0	171
Average running time (s)	2.634	600.212
Success rate	100%	0%