Assignment-4

Machine Data and Learning
Search Trees and Local Search

Due Date: 19 April, 2025 11:59 PM

Important Notes

- Submissions must be handwritten. Typed or digital submissions will not be accepted.

TOTAL: [35 PTS]

1 Question 1: Optimized Robot Path Planner (21 points)

1.1 Scenario:

A robot is placed at the origin point $(0, 0)$ in a grid-based environment. The robot can move in exactly three directions, creating a search tree with a fixed branching factor of 3. The available moves are as follows:

- Right (R): Move to $(x + 1, y)$.
- Up (U): Move to $(x, y + 1)$.
- Diagonal (D): Move to $(x + 1, y + 1)$.

The robot starts at position $(0, 0)$.

Goal: Find any position (x, y) such that:

$$xy - 2x - 3y + 6 = 0$$

1.2 Task:

Generate a tree where each node is a coordinate (x, y), and its children are produced by applying the three movement options.

Task:

1. Generate the search tree up to depth 3 (i.e., 3 moves from the starting point) for each of the following search strategies :
   - Breadth-First Search (BFS)
   - Depth-First Search (DFS)
   - Iterative Deepening Depth-First Search (IDDFS)
2. For each strategy, report the order in which nodes are expanded and indicate the first valid goal node found (if any).
3. Follow the child visitation priority (Right $\rightarrow$ Up $\rightarrow$ Diagonal) strictly for all strategies.

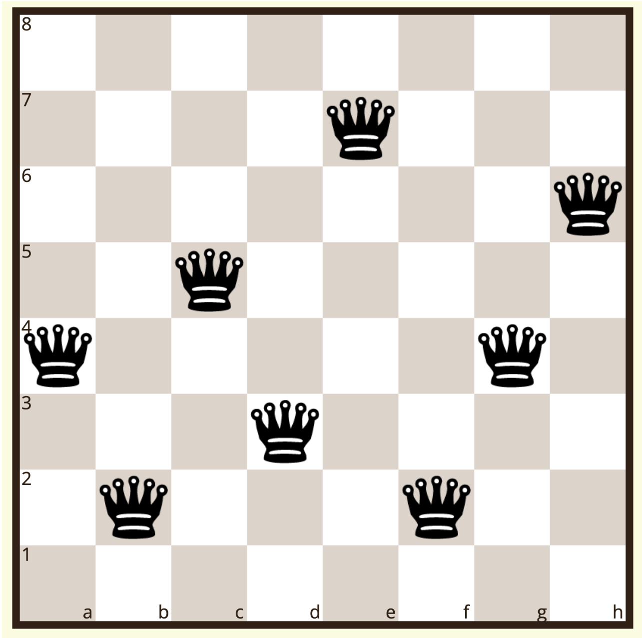

Figure 1: Initial configuration(for Part A) with 4 attacking pairs of queens: $(A4, G4)$, $(B2, F2)$, $(C5, E7)$, and $(C5, F2)$.

2 Question 2: Local Search – Hill Climbing and Beam Search (14 points)

You are provided with an initial configuration of the 8-Queens problem,represented as a board snapshot. Each queen is placed in a unique column, and the snapshot also includes the number of conflicts (i.e., number of attacking pairs of queens) for each position, as typically presented in the textbook. Your task is as follows:

2.1 Part A: Hill Climbing (7 points)

Starting from the given board configuration(Figure 1):

- Perform 3 steps of the hill climbing search algorithm.
- At each step, evaluate the possible moves for each queen within its column.
- Choose the move that leads to the lowest number of total conflicts.
- In case of ties, you may choose any one of the equally good moves.
- Clearly show the board configuration and conflict count after each step.

2.2 Part B: Beam Search (7 points)

Consider a scenario where you begin with 3 different initial board configurations(Figure 2,Figure 3 and Figure 4):

Perform 2 steps of beam search with a beam width of 3.

- At each step, generate all possible successors (one-move variations per queen) of the current beam candidates.
- From the combined pool of successors, select the top 3 configurations with the lowest total conflict values to form the next beam.
- Again, in case of ties, you may choose arbitrarily among equally scoring candidates.
- Show the board states and their conflict counts at each step.

NOTE: Please ensure that your board representations are clear and conflict counts are properly indicated.

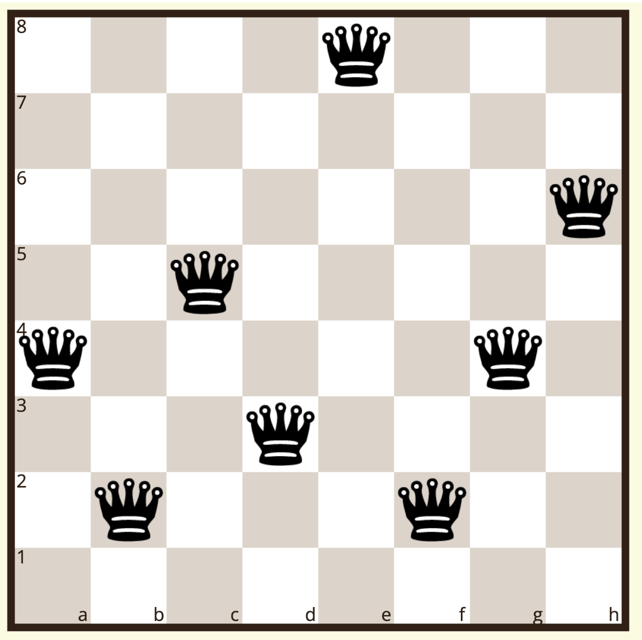

Figure 2: First initial configuration(for Part B) with 4 attacking pairs of queens: $(A4, G4)$, $(A4, E8)$, $(B2, F2)$ and $(C5, F2)$.

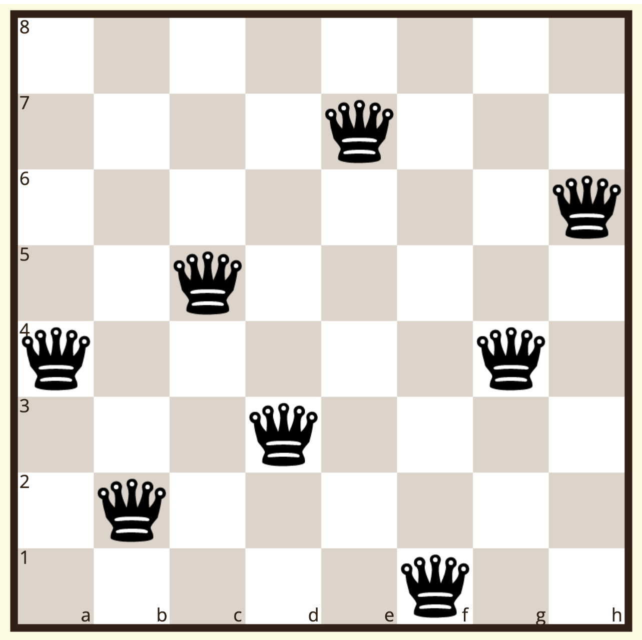

Figure 3: Second initial configuration(for Part B) with three attacking pair of queens: $(A4, G4)$, $(C5, E7)$ and $(D3, F1)$.

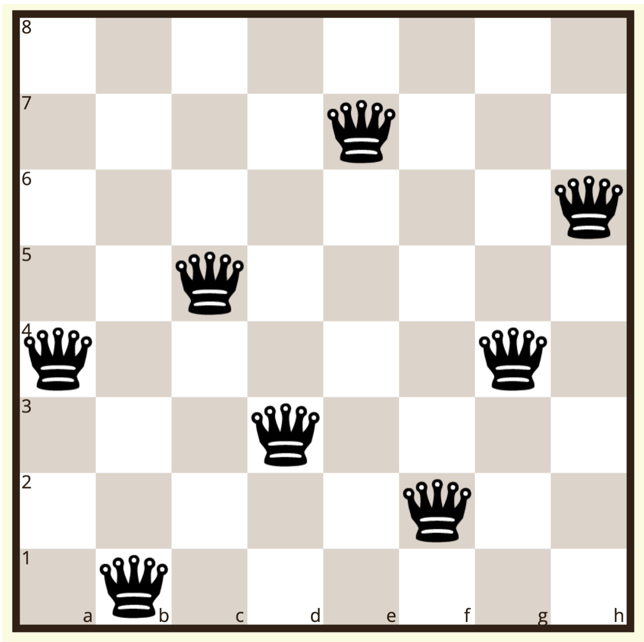

Figure 4: Third initial configuration(for Part B) with five attacking pair of queens: $(A4, G4)$, $(B1, D3)$, $(C5, E7)$, $(C5, F2)$ and $(D3, B1)$.