ARTIFICIAL INTELLIGENCE
ASSIGNMENT-3: Hill Climbing

Name: Abhishek Singh
Roll No.- 2402res24

Introduction

We are tasked with implementing the Hill Climbing Search Algorithm to solve the 8-puzzle problem. The goal is to find the optimal solution by exploring states in the local region, continuously searching for a better solution.

Algorithm Description

- The Hill Climbing Search Algorithm is a local search technique that constantly looks for a better solution in its neighborhood.
- It involves starting with an initial state and iteratively moving to neighboring states that improve the objective function until a local maximum is reached.

Implementation Details

- We implemented the algorithm in Python.
- The code consists of functions and classes to represent the puzzle state, perform the search, and calculate heuristic functions.
- We defined classes for the puzzle state (`EightPuzzleState`), the search algorithm (`HillClimbingSearch`), and the heuristic functions (`O1` and `O2`).
- The algorithm iterates through neighboring states, evaluates them using the selected heuristic function, and moves to the best neighboring state until termination conditions are met.

Output Analysis

We executed the algorithm with different initial states and heuristic functions. The output includes information such as start state, goal state, total states explored, and whether the algorithm succeeded or failed.

Observation and Conclusion

1. **Failure Rate:**
    - The failure rate for both heuristic functions, O1(n) and O2(n), is quite high. Many initial states failed to reach the goal state using hill climbing with both heuristics.

2. **Total States Explored Before Termination:**
    - The total number of states explored before termination varied widely across different initial states and heuristic functions.
    - For some initial states, the number of explored states was relatively low, indicating that hill climbing quickly found a solution. However, for many others, the exploration was extensive before reaching the termination condition.

3. **Effectiveness of Heuristic Functions:**
    - Both heuristic functions, O1(n) and O2(n), were not consistently effective in guiding the search process towards the goal state.
    - In some cases, particularly when using heuristic O2(n), the number of explored states before termination was significantly higher compared to heuristic O1(n).

4. **Comparison of Heuristic Functions:**

- Heuristic O2(n) generally explored more states before termination compared to heuristic O1(n).
- However, in the one successful case reported, heuristic O2(n) was able to find a solution with fewer explored states compared to other failed attempts.

5. **Observation on Successful Case:**
    - In the one successful case reported, heuristic O2(n) found a solution with 31 explored states. This suggests that O2(n) might be more effective in certain scenarios, although further analysis is needed to draw a definitive conclusion.

6. **Impact of Initial State:**
    - The choice of the initial state significantly influenced the effectiveness of the hill climbing algorithm. Some initial states might be inherently more challenging to solve using local search techniques.

7. **Limitations and Areas for Improvement:**
    - The high failure rate and variable performance of the heuristic functions indicate the limitations of hill climbing in solving the 8-puzzle problem.
    - Possible areas for improvement include refining the heuristic functions to better estimate the distance to the goal state and exploring alternative local search algorithms that might offer better performance.

Overall, the analysis suggests that while hill climbing with heuristic functions O1(n) and O2(n) can sometimes find solutions for the 8-puzzle problem, it is not consistently effective.

Final Thoughts

In conclusion, while the Hill Climbing algorithm showed promise, especially in the successful case with heuristic O2(n), it struggled to consistently find solutions for the 8-puzzle problem.