CS250 - ARTIFICIAL INTELLIGENCE LAB

ASSIGNMENT-8: Simulated Annealing & Random Walk

Hill climbing suffers from problems in getting stuck at local minima (or maxima). We could try to overcome these problems by trying various techniques. Some examples are -

• We could try a hill climbing algorithm using different starting points.

• We could increase the size of the neighbourhood so that we consider more of the search space at each move. For example, we could try 3-opt, rather than a 2-opt move when implementing the TSP. Unfortunately, neither of these have proved satisfactory in practice when using a simple hill climbing algorithm.

Simulated annealing solves this problem by allowing worse moves (lesser quality) to be taken some of the time. That is, it allows some uphill steps so that it can escape from local minima.

Unlike hill climbing, simulated annealing chooses a random move from the neighbourhood. If the move is better than its current position then simulated annealing will always take it. If the move is worse (i.e. lesser quality) then it will be accepted based on some probability.

Acceptance Criteria

The probability of accepting a worse state is given by the equation –

$$P = \exp(-c/t) > r$$

where,

c = the change in the evaluation function

t = the current temperature

r = a random number between 0 and 1.

The probability of accepting a worse move is a function of both the temperature of the system and of the change in the cost function.

h1: Number of displaced tiles:

- This heuristic calculates the number of tiles that are not in their goal position. It's admissible for the 8-puzzle problem.

h2: Total Manhattan distance:

- This heuristic calculates the total Manhattan distance (sum of horizontal and vertical distances) of each tile from its goal position. It's also admissible for the 8-puzzle problem.

Constraints to be Checked:

a. Admissibility check:

- Both h1 and h2 are admissible heuristics for the 8-puzzle problem because they never overestimate the cost to reach the goal state.

b. New Heuristic h3= h2*h1:

- The new heuristic h3 can sometimes be admissible if both h1 and h2 are admissible. However, it may not be consistent, which can lead to suboptimal solutions or even failure to find a solution.

c. Considering a blank tile as another tile:

- Considering the blank tile as another tile changes the problem's state space and may affect the admissibility and consistency of heuristics. It may lead to different search paths and potentially different solutions.

d. Handling Local Optimum:

- The starting temperature must be hot enough to allow a move to almost any neighbourhood state. If this is not done then the ending solution will be the same (or very close) to the starting solution. Alternatively, we will simply implement a hill climbing algorithm. However, if the temperature starts at too high a value, then the search can move to any neighbour and thus transform the search (at least in the early stages) into a random search. Effectively, the search will be random until the temperature is cool enough to start acting as a simulated annealing algorithm.

Random Walk scenario

To check whether a random walk scenario occurs in the Simulated Annealing algorithm for solving the 8-puzzle problem, we can examine the behaviour of the algorithm over multiple runs. A random walk scenario occurs when the algorithm makes random moves without converging towards a solution.

Random Walk scenario occurs when the temperature is too high.

Here's how we can analyse it:

1. Multiple Runs: Execute the Simulated Annealing algorithm for solving the 8-puzzle problem multiple times with different initial states.

2. Success Rate: Record the success rate, i.e., the percentage of runs that find a solution.

3. Solution Path Length: For successful runs, analyse the length of the solution path. A random walk scenario is likely if the solution paths are significantly longer than expected.

4. Convergence Behaviour: Monitor the behaviour of the algorithm during each run. If the algorithm frequently accepts worse moves (moves leading to higher heuristic values) even when the temperature is relatively low, it suggests a random walk scenario.

5. Compare Against Baseline: Compare the algorithm's behaviour against a baseline or benchmark. For example, compare it against a deterministic search algorithm like A* search to see if the Simulated Annealing algorithm is significantly less effective.

By running the `random walk analysis` function with a sufficient number of runs, we can assess whether the Simulated Annealing algorithm exhibits a random walk scenario in solving the 8-puzzle problem. If the success rate is low, or the average solution length is unexpectedly high, it may indicate a random walk scenario.