Assignment (1)

Name: Samiha Akter Maisha

Roll no: 0112231010

Section: (D)

Course: Artificial Intelligence

Question no (1)

* Hill Climbing:

Merits: Easy to implement and fast conjunction

Demerits: Easily get stuck in local optima and there is no backtraking.

Handling Local Minima: It is poor cause gets stuck in local minima.

Global Optimality: Unlikely to find global optimum.

Simulated Annealing:

Merits: It can escape local minima using random exploration, balance exploration and exploitation.

Demerits: Slower conjunction and requires tunning of temparatur schedule.

Handling Local minima: It's good because probabilistically escape local minima.

Global Optimality: Moderate may reach global optimum with good tuning.

Beam Search:

Merits: Efficient with large state space and uses multiple candidates.

Demerit: May coverage to suboptimal solution if beam width is small and not guraranted to explore all paths.

Handling Local Minima: It's Medium and depends on the beam width and heuristic quality.

Global Optimality : Moderate better than hill climbing but not guaranted.

Genetic Algorithm :

Merits: It explores large search space, maintains diversity, high adaptibility.

Demerits : Requires careful parameter tuning and slow for every big problems.

Handling Local Minima : It's very good and maintains multiple diverse solution and uses crossover.

Global Optimality: High and high ~~es~~ chance o finding global optimum.

2no question:

Hill climbing:

Efficiency: It's very fast and low memory usages.

Complexity: $O(b^d)$. Here b is branching factor and d is depth.

Scalability: Poor and does'nt scale well. and sometimes performence drops.

Simulated Annealing:

Efficiency: Slower due to probabilistic decisions.

Complexity: $O(n^2) + O(n^3)$ depends on cooling schedule

Scalability: Moderat. It handles larger spaces better than hill climbing.

Beam Search:

Efficiency: Moderate efficiency and parallelizable.

Complexity: $O(k \times b^d)$ - k = beam width.

Scalalbility: Good because it can increase beam width to handle bigger problems.

Genetic Algorithm:

Efficiency: It's computationall expensive

Complexity: $O(g \times p \times n)$ - g = generation, p = population size, n = proble size.

Scalability: Very good and ideal for ~~lag~~ larger, complex problem.

3no ans:

Based on my analysis the most suitable algorithm is Genetic Algorithm.

Justifying my recommendation below:

(1) It handles large solution spaces ~~effiecently~~. efficiently.

(2) Maintains diversity of routes using populations.

(3) Escapes local minima using mutation and crossover.

(4) Can incorporate constraints like delivery time window, traffic etc.

Optimal Combination:

Use simulating annealing to refine the best solution from the Genetic Algorithm to further improve local optimization.