Department of Computer Engineering
Academic Term II: 23-24

Class: B.E (Computer), Sem – VI **Subject Name: Artificial Intelligence**

Student Name: **Roll No:**

Practical No:	**7**
Title:	Block World Problem solving by hill climbing approach
Date of Performance:	18-03-2024
Date of Submission:	25-03-2024

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Marks
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Logic/Algorithm Complexity analysis (03)	03(Correct)	02(Partial)	01 (Tried)	
3	Coding Standards (03): Comments/indention/Naming conventions Test Cases /Output	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Assignment (03)	03(done well)	2 (Partially Correct)	1(submitted)	
Total					

Signature of the Teacher:

Experiment No: 7

Title: Block world problem solving by Hill Climbing method

Objective: To study the solution for block word problem by Hill Climbing approach

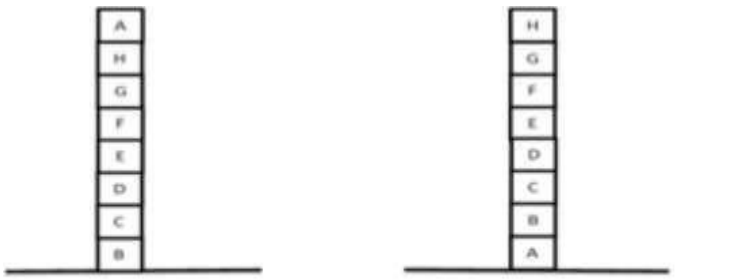
Theory:

SIMPLE HILL CLIMBING

1. Evaluate the initial state. If it is also a goal state, then return it and quit. Otherwise, continue with the initial state as the current state.
2. Loop until a solution is found or until there are no new operators left to apply in the current state:
    a) Select an operator that has not yet been applied to the current state and apply it to produce a new state
    b) Evaluate the new state,

i. If it is a goal state, then return it and quit.
ii. If it is not a goal state but it is better than the current state, then make it the current state.
iii. If it is not better than the current state, then continue in the loop.

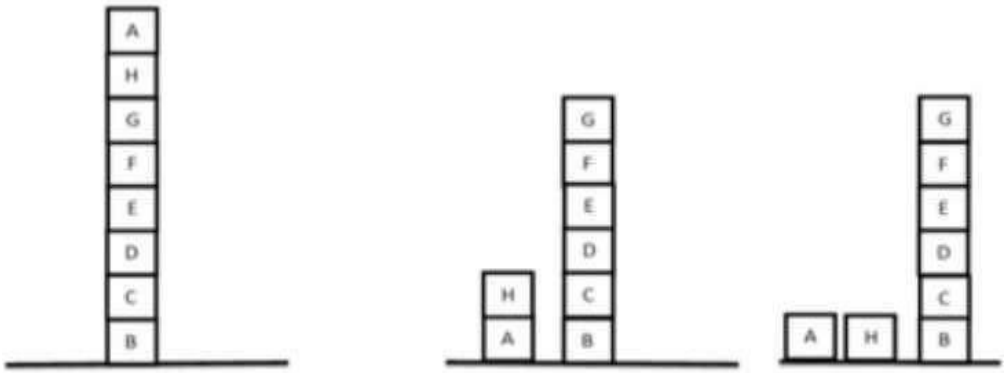
Consider the blocks world problem. Assume the same operators (i.e., pick up one block and put it on the table; pick up one block and put it on another one) suppose it uses the following heuristic function:



HILL CLIMBING PROBLEM

Local: Add one point for every block that is resting on the thing it is supposed to resting on.

Subtract one point for every block that is sitting on the wrong thing.
Using this function, the goal state has a score of 8. The initial state has a score of 4 (since it gets one point added for blocks C, D, E, F, G and H and one point subtracted for blocks A and B).



THREE POSSIBLE STATES

There is only one move from the initial state, namely to move block A to the table. That produces a state with a score of 6. The hill-climbing procedure will accept that move. From the new state, there are three possible moves, leading to the three states.
These states have the score: (a) 4, (b) 4, and (c) 4. Hill climbing will halt because all these states have lower scores than the current state. The process has reached a local maximum that is not the global maximum.

OUTPUT:

Post Lab Questions:

1. What are the advantages and disadvantages of state space search?
2. What are the advantages and disadvantages of the Hill Climbing approach?
3. Describe variations of Hill Climbing approach
4. Solve the Block World problem by using the STRIPS method.

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

9526 TE COMPS A A.I Exp 7

PostLab :-

① What are the advantages and disadvantages of state space search?

⇒ Advantages:

1. Completeness : Guaranteed to find a solution if one exists
2. Optimally : can find the best solution
3. Flexibility : Adoptable to various problem domains.
4. Applicability: Suitable for a wide range of problems.

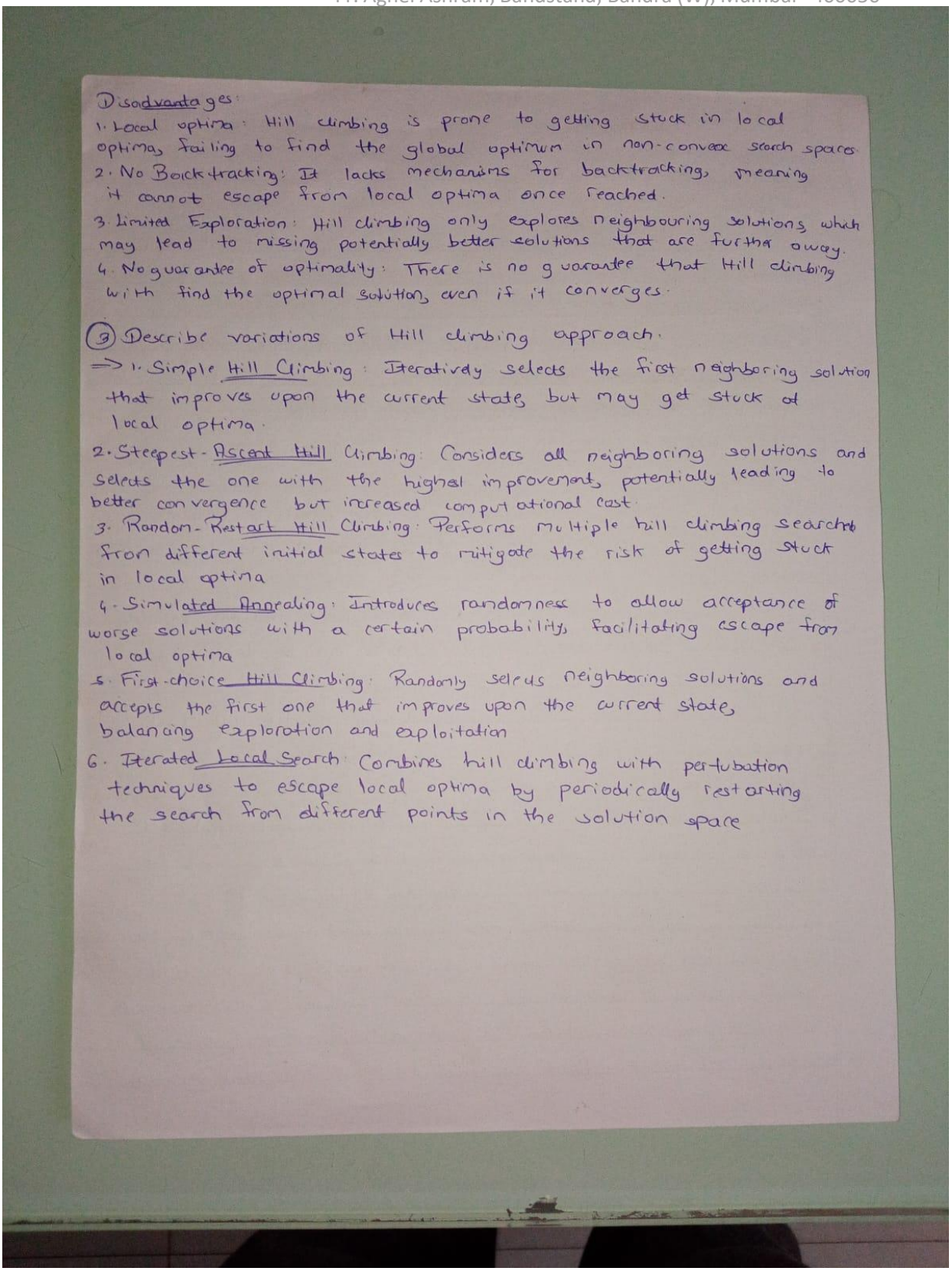
Disadvantages:

1. Exponential Complexity: Computationally expensive for large problems
2. Memory Requirements: High memory usage, especially for large search spaces.
3. Heuristic Dependency : Effectiveness relies heavily on heuristic equality.
4. Optimization Challenges: finding optimal solutions can be difficult

② What are advantages and disadvantages of the Hill climbing approach?

⇒ Advantages:

1. Simplicity: Hill climbing is easy to understand and implement, making it suitable for simple optimization problems
2. Efficiency: It can converge quickly to a local optimum, especially in problems with smooth and continuous search spaces.
3. Memory Efficiency: Hill climbing typically requires minimal memory, making it suitable for recource-constrained environments.
4. Low computational overhead: It involves minimal computational overheads, making it efficient for real-time or embedded systems.



Disadvantages:
1. Local optima: Hill climbing is prone to getting stuck in local optima, failing to find the global optimum in non-convex search spaces.
2. No Backtracking: It lacks mechanisms for backtracking, meaning it cannot escape from local optima once reached.
3. Limited Exploration: Hill climbing only explores neighbouring solutions, which may lead to missing potentially better solutions that are further away.
4. No guarantee of optimality: There is no guarantee that Hill climbing will find the optimal solution, even if it converges.

(3) Describe variations of Hill climbing approach.
=> 1. Simple Hill Climbing: Iteratively selects the first neighboring solution that improves upon the current state, but may get stuck at local optima.
2. Steepest-Ascent Hill Climbing: Considers all neighboring solutions and selects the one with the highest improvement, potentially leading to better convergence but increased computational cost.
3. Random-Restart Hill Climbing: Performs multiple hill climbing searches from different initial states to mitigate the risk of getting stuck in local optima
4. Simulated Annealing: Introduces randomness to allow acceptance of worse solutions with a certain probability, facilitating escape from local optima
5. First-choice Hill Climbing: Randomly selects neighboring solutions and accepts the first one that improves upon the current state, balancing exploration and exploitation
6. Iterated Local Search: Combines hill climbing with perturbation techniques to escape local optima by periodically restarting the search from different points in the solution space

(4) Solve the Block World problem by using the STRIPS method.

→ Block World problem using the STRIPS (Stanford Research Institute Problem Solver)

Initial State: On (A, Table)
On (B, Table)
On (C, B)

Goal State: On (A, B)
On (B, Table)
On (C, Table)

Operators:-

1. Pick up (block, source)
Precondition: On (block, source), clear (block).
Effects: ¬ on (block, source), Holding (block)

2. Putdown (block, destination)
Precondition: Holding (block)
Effects: On (block, destination), clear (block), ¬Holding (block)

3. Stack (block, source, destination)
Precondition: Holding (block), clear (destination)
Effects: ¬on (block, source), On (block, destination), clear (source), ¬Holding (block)

4. Unstack (block, source, destination)
Preconditions: On (block, source), clear (block), clear (destination)
Effects: On (block, source), clear (destination), Holding (block)