ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

HEURISTIC SEARCH, HILL CLIMBING

Session – 8

AIM OF THE SESSION

To familiarize students with the basic concept of Heuristic functions

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Explain What is Local Search Algorithm?
2. Describe different types of Local Search Algorithm:Hill Climbing Algorithm, Simulated Annealing

LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Define the concept of Local Search Algorithm
2. Understand about types of Local search, Hill Climbing Algorithm, Simulated Annealing

SESSION INTRODUCTION

- This session provides the basics of Heuristic Search, Hill Climbing, and Simulated Annealing.
- This session also gives an idea about optimization.

Heuristic Search Algorithm

What Is Heuristic Search In AI?

- A **heuristic search algorithm** is a problem-solving method that uses a **heuristic function**—a rule of thumb or educated guess—to guide the search for solutions in complex spaces, especially when finding an exact answer would be too slow or computationally expensive.
- In artificial intelligence for tasks like pathfinding, game playing, and optimization problems used.
- The heuristic function provides an estimate (not always exact) of the cost or value of a particular state, helping the algorithm decide which direction to search.
- **Heuristic search algorithms** include **A*, Greedy Best-First Search, Hill Climbing**, and **Simulated Annealing.**

Heuristic Search Algorithm

HILL CLIMBING

- Artificial intelligence (AI) uses the straightforward optimization method known as **"hill climbing" to identify the optimum answer to a given issue**.
- **Hill Climbing** is a heuristic optimization process that iteratively advances towards a better solution at each step in order to find the best solution in a given search space.
- Types are
  - **Simple hill climbing**
  - **Steepest-ascent hill climbing**
  - **Stochastic hill climbing**

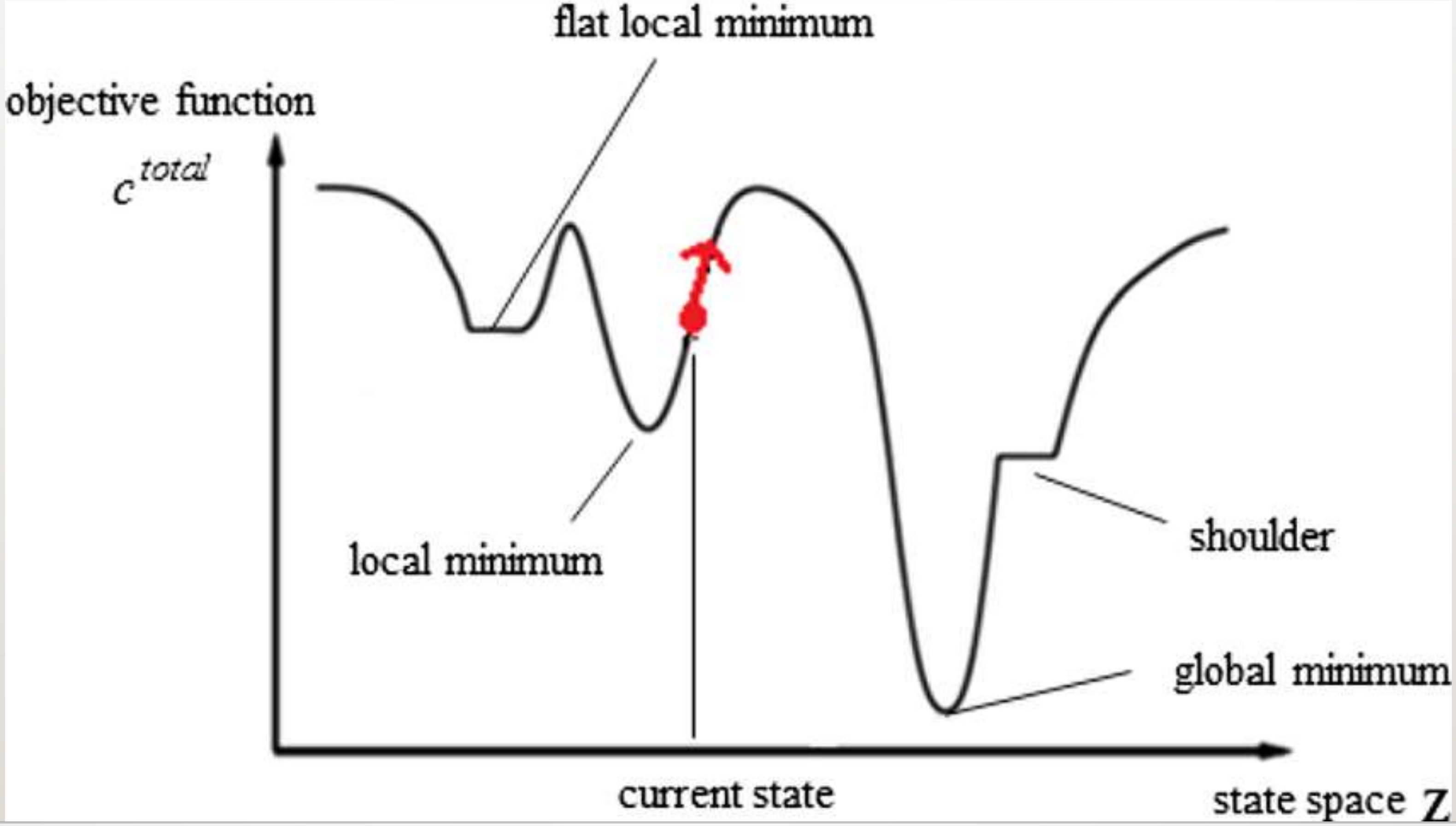
TYPES OF HILL CLIMBING

- **Simple hill climbing:**
  - This examines one neighboring node at a time and selects the first one that optimizes the current cost to be the next node..
- **Steepest-ascent hill climbing:**
  - This examines all neighboring nodes and selects the one closest to the solution state.
- . **Stochastic hill climbing:**
  - This selects a neighboring node at random and decides whether to move to it or examine another.

ALGORITHM FOR SIMPLE HILL CLIMBING

1. Evaluate initial state- if goal state, stop and return success. Else, make initial state current.
2. Loop until the solution reached or until no new operators left to apply to current state:
   - a. Select new operator to apply to the current producing new state.
   - b. Evaluate new state:
     - If a goal state, stop and return success.
     - If better than the current state, make it current state, proceed.
     - Even if not better than the current state, continue until the solution reached.
3. Exit.

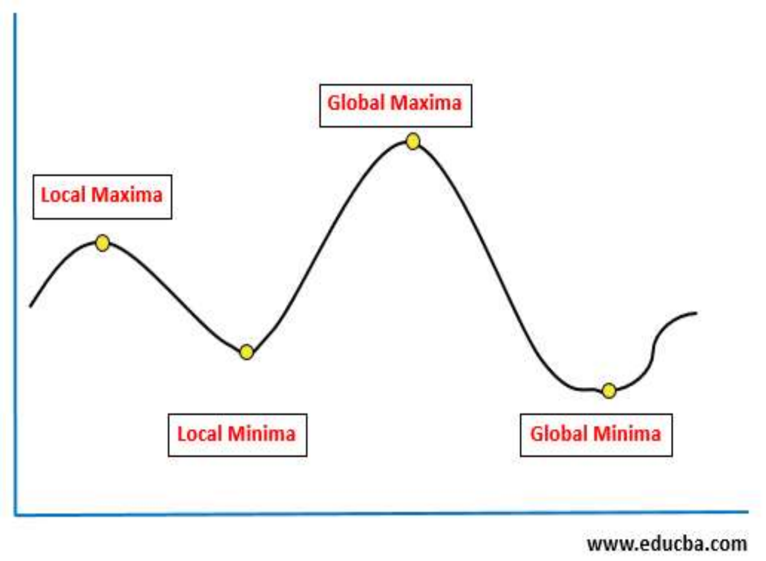
STATE SPACE DIAGRAM FOR HILL CLIMBING



PROBLEMS IN HILL CLIMBING

- **Local Maxima:** A local maxima is a state that dominates its surrounding states but is surpassed by another state that is also present and is higher than the local maximum.

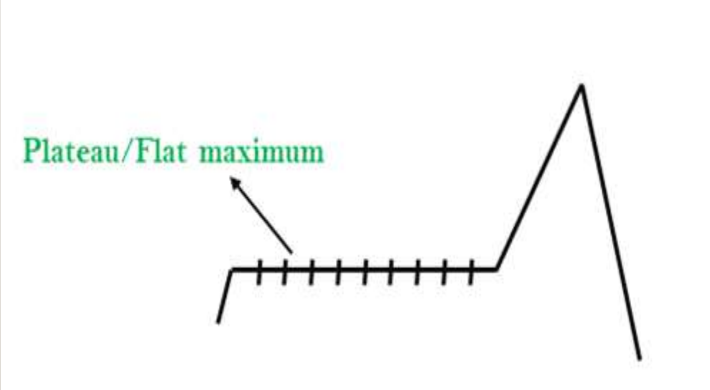
  Solution: In the state space landscape, the local maximum can be resolved using the backtracking approach. Make a list of the promising paths so that the algorithm may go back and investigate other routes.



PROBLEMS IN HILL CLIMBING

Plateau: A plateau is a flat region of the search space where all neighbor states of the current state have the same value. As a result, the algorithm is unable to determine the optimum course of action. In the plateau region, a hill-climbing search may become disoriented.

Solution: Making either large or little movements while looking for a solution can help you get over the plateau. Choose a state at random that is remote from the one you are in now, giving the algorithm a chance to discover a non-plateau region.

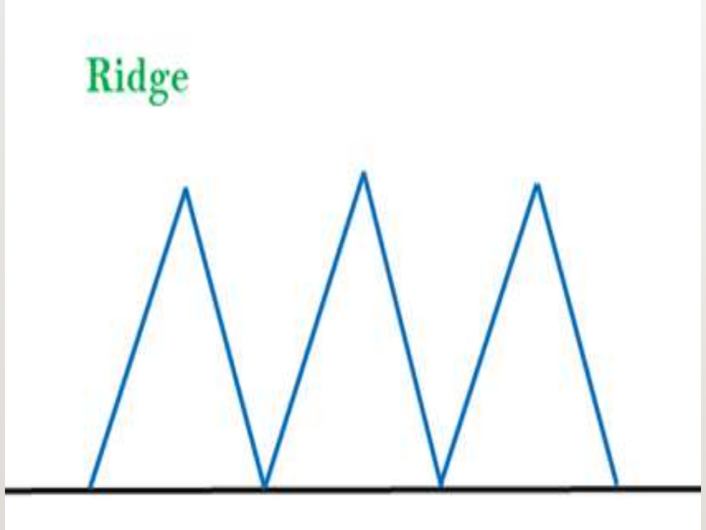


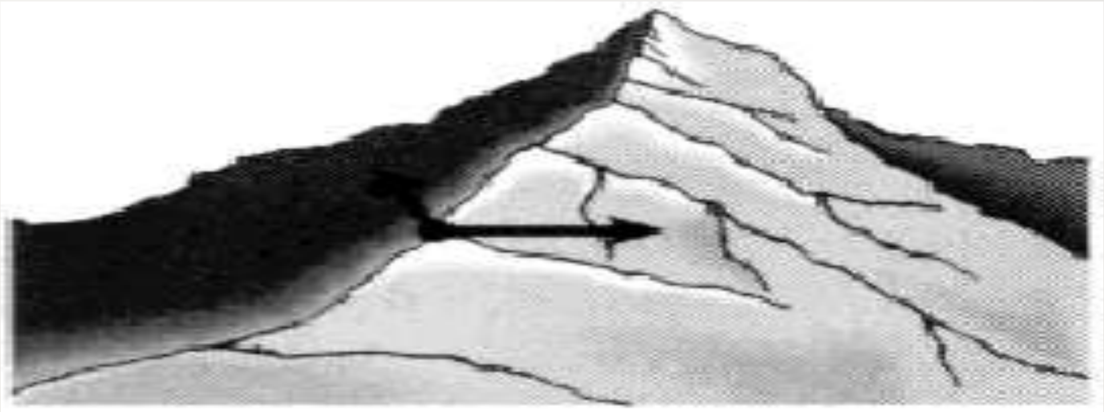
KL UNIVERSITY

PROBLEMS IN HILL CLIMBING

Ridges: The local maximum might take distinctive forms, such as ridges. It has a portion that is higher than the places around it, but because of its own slope, it cannot be reached in a single motion.

Solution: We can make this situation better by using bidirectional search or by going in separate ways.



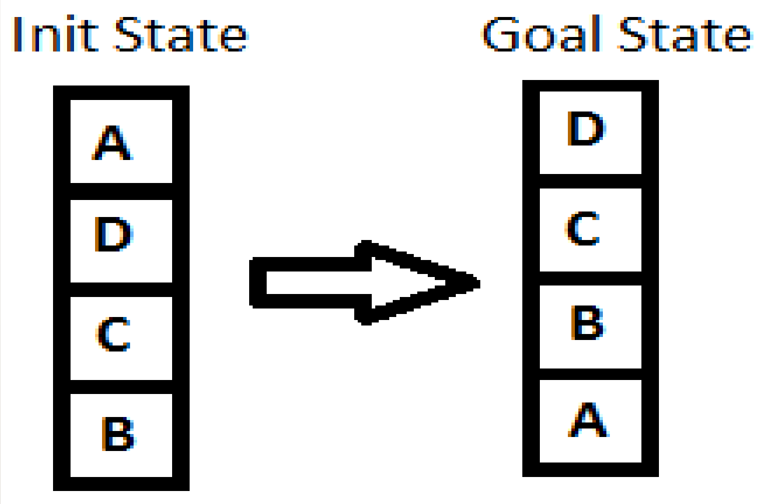


BLOCKS WORLD PROBLEM

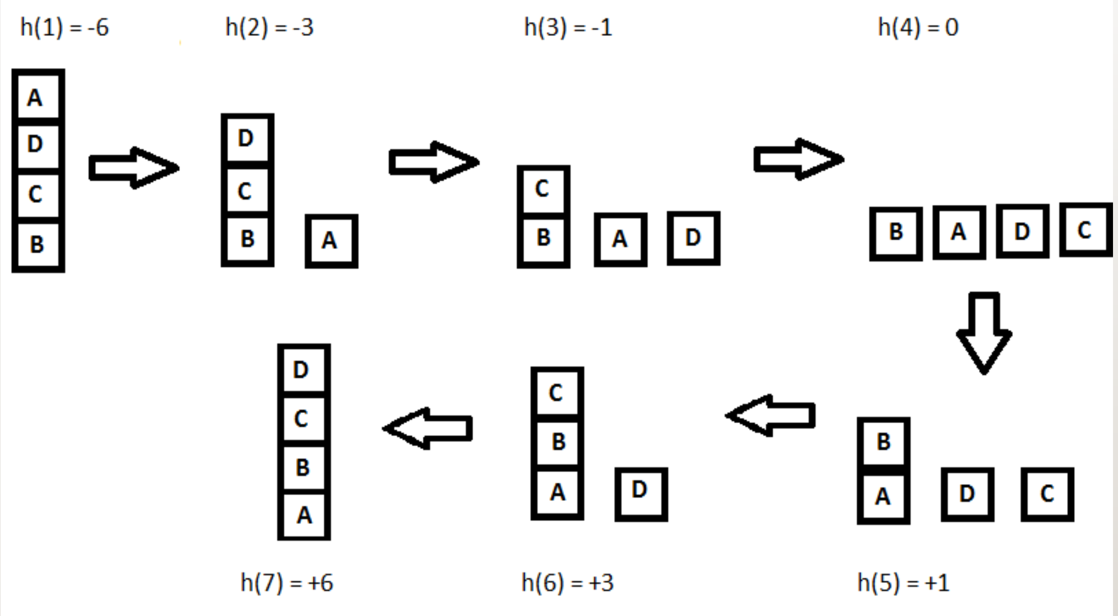
- The most important step in solving any hill-climbing issue is to pick the right heuristic function.

Definition of such a function, h

- If the block is appropriately positioned, h(x) = +1 for all the blocks in the support structure; otherwise, h(x) = -1 for all the blocks in the support structure.



- Any block that has the same support structure as the objective state is referred to in this context as being appropriately positioned. Let's have a look at all the iterations and their heuristics to achieve the goal state using the hill climbing approach we outlined earlier:



Example:
https://www.youtube.com/watch?v=2SIO34_VsY4

STEEPEST-ASCENT HILL CLIMBING

This is a variation of simple hill climbing the algorithm selects a neighboring solution at random and decides whether to move to that neighbor based on its improvement over the current solution. Also known as Gradient Ascent search.

Algorithm: **Example:** https://www.youtube.com/watch?v=pGrGfmgDZtw

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 a complete iteration produces no change to the current state
   a. Let SUCC be a state such that any possible successor of the current state will be better than SUCC
   b. For each operator that applies to the current state do:
      i. Apply the operator and generate a new state
      ii. Evaluate the new state. If is a goal state, then return it and quit. If not, compare it to SUCC. If it is better, then set SUCC to this state. If it is not better, leave SUCC alone.
   c. If the SUCC is better than the current state, then set the current state to SUCC.

SIMULATED ANNEALING

- Simulated Annealing is a modified version of stochastic hill climbing.
- It is a heuristic search algorithm applied to achieve optimization in artificial intelligence issues.
- It explores the search space and avoids local optimum by employing a probabilistic method to accept a worse solution with a given probability.
- In simulated annealing, "temperature" is a parameter that controls the probability of accepting worse solutions as the algorithm progresses. It is analogous to the temperature in the physical annealing process where materials are heated and then slowly cooled to remove defects and find a low-energy state.

Example:
https://www.youtube.com/watch?v=uK6nbn7hArQ

SIMULATED ANNEALING

Key Points about Temperature in Simulated Annealing:

- **High Temperature**: At the start, the temperature is set high, allowing the algorithm to accept worse solutions with higher probability. This helps in exploring the solution space more thoroughly and avoids getting trapped in local optima.
- **Cooling Schedule**: The temperature is gradually reduced according to a cooling schedule. This schedule dictates how quickly the temperature decreases over time.
- **Low Temperature**: As the temperature decreases, the probability of accepting worse solutions decreases, allowing the algorithm to converge to an optimal or near-optimal solution.

STEPS OF THE SIMULATED ANNEALING ALGORITHM

- Start with an initial solution.
- Set the initial temperature to a high value.
- Repeat the following steps until the stopping criterion is met:
  - Generate a new solution by making a small modification to the current solution.
  - Evaluate the objective function of the new solution.
  - If the new solution improves the objective function, accept it as the new current solution.
  - If the new solution does not improve the objective function, accept it with a probability that depends on the difference between the objective function values of the current and new solutions and the current temperature.
  - Decrease the temperature according to a cooling schedule.

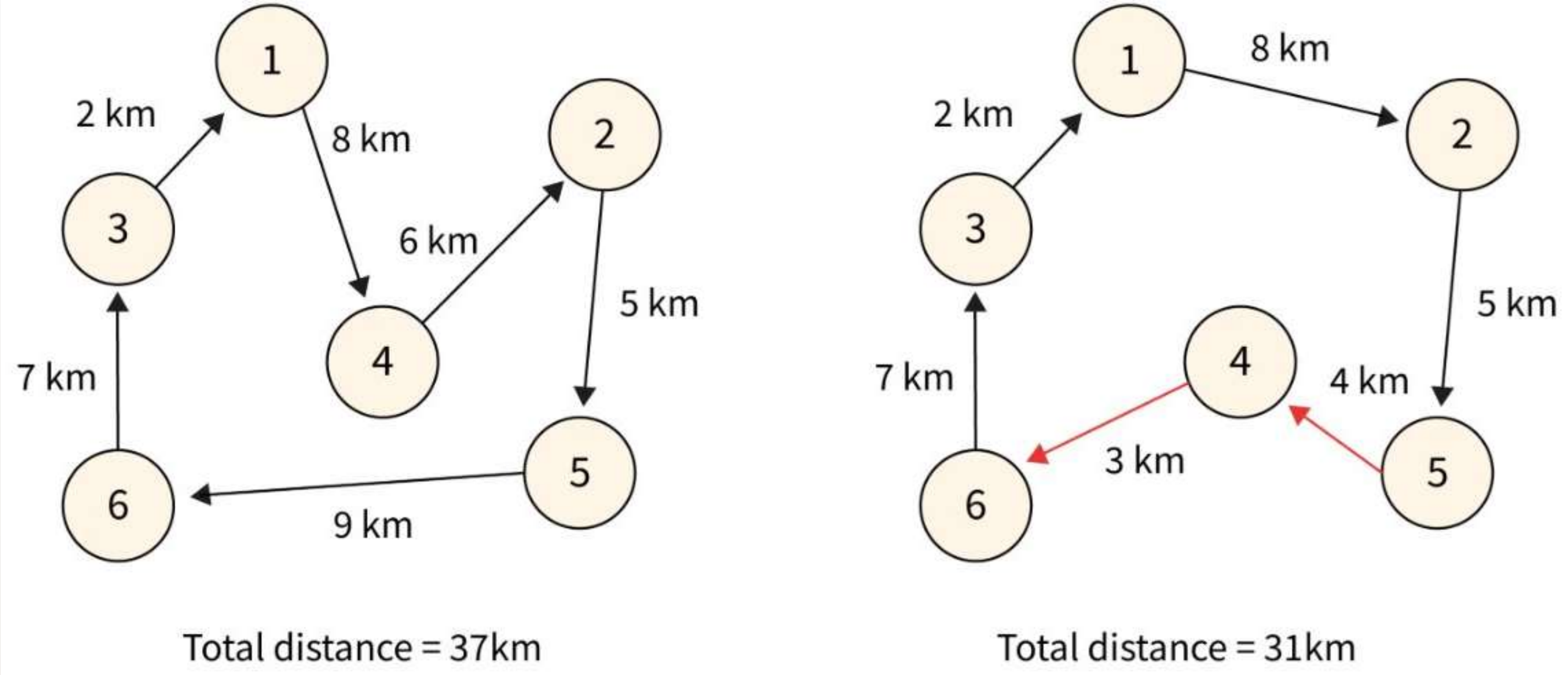
STEPS OF THE SIMULATED ANNEALING ALGORITHM

- Return the current solution as the final solution.
- The main principle of the simulated annealing algorithm is to control the level of randomness in the search process by altering the temperature parameter.
- High temperatures enable the algorithm to explore new regions of the search space by increasing its propensity to accept non-improving moves. The algorithm becomes more selective and concentrates on improving the solution as the temperature drops.
- Simulated annealing has been successfully applied to a wide range of optimization problems, such as the traveling salesman problem, the vehicle routing problem, and the job shop scheduling problem. However, it requires careful tuning of the temperature and cooling schedule parameters to achieve good performance.

TRAVELLING SALESMAN PROBLEM

- The Traveling Salesman Problem (TSP) is a well-known example of a combinatorial optimization problem in which the goal is to determine the quickest path between the starting city and a given set of cities. No known algorithm can complete it in polynomial time because it is an NP-hard problem.
- **Local search algorithms for the TSP work** by starting with an initial solution and incrementally improving it by making small changes to it one at a time until no more advancements are possible.

TRAVELLING SALESMAN PROBLEM



SUMMARY

Local search algorithms iteratively improve a single solution by making small modifications.

- Local search algorithms are often simple and efficient and can handle large solution spaces.
- However, they are prone to get stuck in local optima, where the solution is not the global optimum.
- To overcome this limitation, various modifications to local search algorithms have been proposed, such as tabu search, simulated annealing, and genetic algorithms.

SELF ASSESSMENT QUESTIONS

1. Local search algorithms can solve?

(a) Any problems
(b) optimization problems
(c) Algorithm

2. Hill climbing is sometimes called.?

(a) Local maxima
(b) greedy local search
(c) optimal solution

TERMINAL QUESTIONS

1. Define the Hill Climbing Algorithm with a suitable example.
2. Write the problems in Hill Climbing Algorithm.
3. Define Ridges and Plateau.
4. List the different regions of the Hill Climbing Algorithm.
5. Write an algorithm of hill climbing heuristic search technique.

Reference Books:

1. Russel and Norvig, 'Artificial Intelligence', third edition, Pearson Education, PHI, (2015)

2. Elaine Rich & Kevin Knight, 'Artificial Intelligence', 3nd Edition, Tata Mc Graw Hill Edition, Reprint(2008)

Sites and Web links:

1. https://www.virtusa.com/digital-themes/heuristic-search-techniques
2. https://towardsdatascience.com/a-star-a-search-algorithm-eb495fb156bb
3. https://www.tutorialandexample.com/local-search-algorithms-and-optimization-problem/
4. http://cgi.di.uoa.gr/~ys02/siteAI2008/local-search-2spp.pdf
5. http://aima.eecs.berkeley.edu/slides-pdf/chapter04b.pdf