AI 2002 – Artificial Intelligence

Practice Questions

(Local Beam and Simulated Annealing)

QUESTION 1:

In this question we are going to pose the subset sum problem of described in first question as an optimization problem and then use Hill climbing strategy (i.e. a local search algorithm) to solve it.

Once again assume that for a set having n elements, a solution is coded using a bit string of length n with a bit being set to 1 if the element is part of the subset and 0 otherwise. Further, assume that the optimality of a solution is computed using 1/ (|S - ∑ | + 1) where S is the required value of sum and ∑ is sum of the subset and | x | represents absolute value of x.

A simple operator to generate a new solution from an existing solution can be defined as follows

NEW_SOLUTION(X) = FLIP A BIT IN THE SOLUTION X

This is equivalent to including an element in the subset or excluding an already chosen element from the subset. It is obvious that for a set of size n we can generate n new solutions from an existing solution.

Use the above operator for generating new solutions along with the hill climbing search strategy (also known as local search) to find a solution for the following subset sum problem.

Find a subset of the set {2, 3, 4, 8, 16} having sum 17. Take the solution 00000 as the starting solution in your local search.

You must show all intermediate steps in the form of the following table. For each iteration show all intermediate solutions considered/generated and the solution selected at that iteration.

Iteration No	Intermediate Solutions	Selected Solution
1	10000(fitness = 1/16) 01000(fitness = 1/15) 00100(fitness = 1/14) 00010(fitness = 1/10) 00001(fitness = 1/2)	00001
2	10001(fitness = 1/2) 01001(fitness = 1/3) 00101(fitness = 1/4) 00011(fitness = 1/8) 00000(fitness = 1/17)	No Better Solution

WE STOP HERE AND THE FINAL SOLUTION IS 00001 corresponding to the subset {16}.

QUESTION 2:

Provide short answers (1-3 sentences) for each of the following questions.

a) **Getting stuck in local minima is a problem of Hill climbing algorithm. Suggest one solution to avoid this problem.**

   Random Restart Hill Climbing is a variant of Hill Climbing in which you can start from a random state if you get stuck in local minima/maxima until you reach global minima/maxima.

b) **State two major difference between Hill-climbing search and Best-first search.**

   1) **Completeness:** Hill-Climbing may get stuck in local maxima/minima and might fail to find a solution if it gets trapped. BFS is complete and it is guaranteed to find a solution if one exists, provided that the search space is finite.
   2) **Search Strategy:** Hill-climbing search iteratively moves to neighboring states to improve the current state and it focuses on finding a local optimum. BFS explores the most promising states first based on a heuristic evaluation function and it focuses on finding a globally optimal solution.

c) **What algorithm would result as a special case if local beam search is applied with k = 1.**

   It would result in a special case known as Steepest-Ascent Hill Climbing. In this case, only one successor is considered at each iteration, and the successor with the highest heuristic value (i.e., the steepest ascent) is chosen to replace the current state. This approach keeps climbing up the gradient of the heuristic function until a local maximum is reached, without considering multiple successors simultaneously.

QUESTION 3:

Given a list of n cities {C1, C2, ... , Cn} a route is a sequence of n distinct cities Ci1, Ci2, ... Cin starting at city Ci1 going to city Ci2 and ending at city Cin. Cost of a route is sum of the costs between successive cities and finally cost of coming back from city Cin to city Ci1.

The travelling salesman problem (TSP) asks the following question:

Given a list of n cities {C1, C2, ... , Cn} and the distances between each pair of cities, what is the shortest possible route that starts at some city, visits each city exactly once and returns to the origin city?

In this question we are going to use Hill climbing strategy (i.e. a local search algorithm) to find a optimal/sub-optimal solution of a TSP problem.

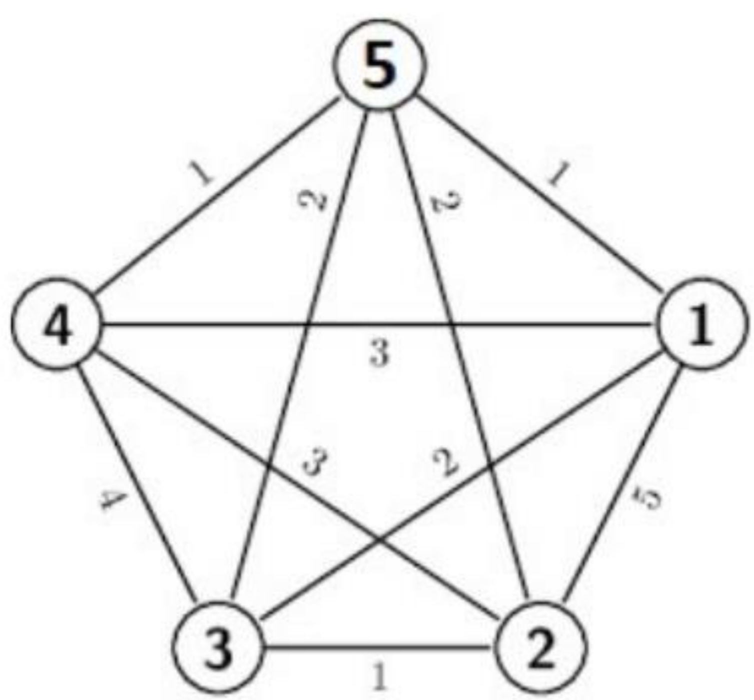
a) **Determine the number of possible routes for a Travelling Salesman Problem (TSP) with n cities, where the objective is to start at some city, visit each city exactly once, and return to the origin city. Provide a concise explanation for your answer.**

The number of possible routes for a TSP problem with n cities can be determined using permutations. Once a starting city is fixed, there are n−1 remaining cities to visit. Therefore, the number of distinct routes is given by (n−1)!, as we can arrange the n−1 remaining cities in (n−1)! ways. This is because permutations account for all possible orderings of the cities, ensuring that each city is visited exactly once before returning to the starting city.

b) **How many new solutions can be generated from an existing solution using a simple operator that involves randomly swapping the positions of two cities in the sequence? Provide reasoning for your answer.**

When using a simple operator that involves randomly swapping the positions of two cities in the sequence, the number of new solutions that can be generated from an existing solution is given by [n x (n - 1)] / 2. This is because there are n choices for the first city to swap and n-1 choices for the second city, resulting in [n x (n - 1)] possible swaps. However, the order of swapping does not matter, so we divide by 2 to avoid counting the same pair twice.

c) **Using the Hill Climbing algorithm with the given method of generating successors, find an optimal solution for the following TSP problem. The problem consists of five cities, and the distances between each pair of cities are provided. The initial solution is randomly generated as 1, 3, 4, 2, 5. Show all intermediate steps, including intermediate solutions and the selected solution at each iteration, along with its cost.**



Steps	Intermediate Solution	Cost
1	1, 3, 4, 2, 5	11
2	1, 4, 3, 2, 5	10
3	3, 4, 1, 2, 5	13
4	3, 4, 1, 5, 2	10
5	3, 5, 1, 4, 2	09

QUESTION 4:

a) **Explain the concept of Simulated Annealing in optimization algorithms. How does it emulate the annealing process in metallurgy, and how is temperature (T) used in the algorithm to control the probability of accepting bad moves? Provide a brief explanation with an example.**

Simulated Annealing emulates the annealing process in metallurgy by gradually reducing the "temperature" parameter (T) to control the probability of accepting "bad moves," analogous to the cooling of a material. At high temperatures, the algorithm is more likely to accept bad moves, allowing exploration of the solution space. As T decreases, the algorithm becomes more selective, focusing on refining the solution towards the optimal state. For example, in a traveling salesman problem, higher temperatures enable the algorithm to explore different routes, while lower temperatures converge towards the shortest route.

b) **In Simulated Annealing, the probability of accepting a move from a state with energy E1 to a state with energy E2 is given by the equation:**

$$p = e^{\frac{(E_2 - E_1)}{kT}} = e^{\frac{-(E_1 - E_2)}{kT}}$$

where e denotes the exponential function, k is Boltzmann's constant, and T is the temperature parameter. If E1 = 10, E2 = 5, and T = 2, calculate the probability of accepting this move. Show your calculations.

The probability of accepting a move from a state with energy E1 to a state with energy E2 in Simulated Annealing is given by the equation:

$$P = e^{-\frac{E_2 - E_1}{kT}}$$

Given E1 = 10, E2 = 5, T = 2, the probability of accepting this move is calculated as follows:

$$P = e^{-\frac{5-10}{2k}}$$

$$P = e^{5/2k}$$

Assuming K = 1 (Boltzmann's constant):

$$P = e^{5/2}$$

$$P \approx 0.0821$$

c) **Discuss the role of the temperature parameter (T) in Simulated Annealing. How does the choice of temperature affect the exploration of the solution space? Provide examples to illustrate how adjusting the temperature impacts the algorithm's performance.**

The temperature parameter (T) in Simulated Annealing balances exploration and exploitation. Higher temperatures promote exploration by accepting bad moves more frequently, while lower temperatures encourage exploitation by focusing on refining the current solution. Adjusting the temperature affects the algorithm's exploration of the solution space, with higher temperatures allowing broader exploration and lower temperatures refining the solution towards optimality.

d) **Explain the concept of convergence in Simulated Annealing. How does the temperature schedule influence the convergence behavior of the algorithm? Discuss any theoretical guarantees regarding convergence and the factors that can affect convergence speed.**

Convergence in Simulated Annealing refers to the algorithm's approach to the global optimum solution. The temperature schedule, which dictates how T decreases over time, influences convergence behavior. A slower decrease in temperature promotes exploration, increasing the likelihood of finding the global optimum, while a faster decrease focuses on exploitation. Theoretical guarantees depend on the temperature schedule and problem characteristics, with appropriately designed schedules leading to convergence to the global optimum with high probability.