Experiment No : 5

Aim : Implement Genetic algorithm using hill climbing

Theory:

**1. Hill Climbing Algorithm**

**Hill Climbing** is a local search optimization algorithm that starts with an arbitrary solution and tries to iteratively improve it by making small changes. The algorithm looks at neighboring solutions and moves to the one with a better objective value. It continues this process until no improvement can be made, i.e., it reaches a local optimum or the global optimum.

**Key Features of Hill Climbing:**

* **Local Search**: It only explores the neighbors of the current solution.
* **Greedy**: It chooses the neighbor that looks best (the highest fitness value), even if it might be a local optimum.
* **No Backtracking**: Once the algorithm moves to a neighbor, it doesn't return to previous solutions.

2. **Genetic Algorithm (GA)**

A **Genetic Algorithm (GA)** is a search heuristic inspired by the process of natural selection. It is a type of evolutionary algorithm where individuals (solutions) evolve through generations to improve their fitness.

**Key Features of Genetic Algorithm:**

1. Population-based: It starts with a population of potential solutions (chromosomes).
2. Selection: Selects the fittest individuals to reproduce.
3. Crossover: Combines parts of two individuals (parents) to produce offspring (new solutions).
4. Mutation: Introduces small random changes to offspring to maintain genetic diversity.
5. Iteration: This process is repeated over multiple generations.

**Steps in a Genetic Algorithm:**

1. Initialization: Start with a random population of solutions.
2. Selection: Select individuals based on their fitness (higher fitness individuals are more likely to be selected).
3. Crossover: Combine two selected individuals to produce offspring.
4. Mutation: Randomly change parts of the offspring to maintain diversity.
5. Replacement: Replace the old population with the new one.
6. Termination: Repeat the process until a stopping criterion is met (e.g., maximum generations or desired fitness level).

To clarify, Hill Climbing is a local search algorithm, and it can be combined with a Genetic Algorithm (GA) to refine solutions through mutation and selection strategies. However, traditionally, Genetic Algorithms involve crossover (mating) and mutation operations, while Hill Climbing is more about local search optimization.

To combine these two concepts, we can use the Hill Climbing method in a population of individuals generated by the Genetic Algorithm. Here's how you can approach it:

1. Population Initialization: Start with a random population of individuals.
2. Fitness Function: Define a fitness function to evaluate the solutions.
3. Selection: Select individuals from the population based on their fitness.
4. Hill Climbing for Refinement: Apply Hill Climbing to improve the selected individuals locally by making small changes to them.
5. Crossover and Mutation: Use Genetic Algorithm operations like crossover and mutation to create new offspring.
6. Termination: Terminate after a certain number of iterations or when convergence is achieved.

import random

def fitness(individual):

return sum(individual)

def hill\_climb(individual):

best = individual

best\_fitness = fitness(best)

for i in range(len(individual)):

neighbor = best[:]

neighbor[i] = random.randint(0, 1)

neighbor\_fitness = fitness(neighbor)

if neighbor\_fitness > best\_fitness:

best = neighbor

best\_fitness = neighbor\_fitness

return best

def genetic\_algorithm\_with\_hill\_climbing(pop\_size, generations, gene\_length, mutation\_rate):

population = [[random.randint(0, 1) for \_ in range(gene\_length)] for \_ in range(pop\_size)]

for generation in range(generations):

population = [hill\_climb(ind) for ind in population]

op :

