LAB-05

Simulation of Annealing.

Objective function : $x^2 + 5\sin x$.

Step 1: define a function called "Simulation-Annealing".

```
def SimulationAnnealing (initial state,
                         initial temp, cooling state,):
    current = initial state.
    best    = current
    best    = objective(current)
    temp    = initial Temp.
    while Temp > 1
        for i ← 1 to it :
            new = neighbour (current)
            current = objective(current)
            new-cost = object(new),
            if function(curr, new-cost,
                temp) > Rand(0,1):
                current = new.
            if new < best:
                best = new
            temp * = cooling
    return(best, best-cost)
```

Step 2: Now define a objective function to change the state

```
def objective(state):
    cost = 0
    for element in state:
        cost += ele**2 + sin(ele)
    return cost
```

Step 3: Next function is to check/search for neighbours

```
def neighbour(state):
    new = state.copy()
    index = rand(0, len(state)-1))
    new[index] += Rand(-1,1)
    return new.
```

Step 4: A function for acceptance probability

```
def AP(current_cost, new_cost, temp)
    if (new_cost < curr_cost):
        return 1
    else:
        return e^((new_cost - curr_cost)/T.
```

Code:

```
def main():
    initial_temp = 1000
    cooling_rate = 0.9
    iterations = 1000
    initial_state = (random.uniform(-10,
```

for _ in range(5):

best_state, best_cost = simulated_annealing(initial_state, initial_temp, cooling_rate, iterations)
print(f"Best state: {best_state}")
print(f"Best cost: {best_cost}")

if __name__ == "__main__":
main()

output:

Best State: [-0.511966, -0.502446, -0.327961, -0.374036, -0.359768]
Best cost: -1.119553