22/10/2024 Lab 5

Stimulated Annealing.

Algorithm.

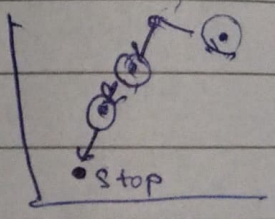
```
function stimulated_Annealing(initial_state,
    initial_temp, cooling_rate, max_iterations)
    stop_state = 1
    current_state = initial_state
    best_state = current_state
    best_cost = obj_func(current_state)
    temp = initial_temp

    while temp > stop_state:
        for i <- 1 to max_iterations:
            new_state = current_state +
                        random(-1, 1)

            cur_cost = obj_func(current_state)
            new_cost = obj_func(new_state)

            if acceptance_prob(cur_cost, new_cost,
                               temp) >
                               random(0,1):
                curstate = new_state

            if new_cost < best_cost:
                best_state = new_state
                best_cost = new_cost

        temp *= cooling_rate

    return (best_state, best_cost)
```

```
function objective_function(state)
    cost = 0
    for ele in state:
        cost += 3*ele**3 + 4
    return cost
```

```
function Acceptance_probability(cur_cost, new_cost, temp)
    if (new_cost < cur_cost):
        return 1
    else
        return e^(-(cur_cost - new_cost)/temp)
```

Objective function

$3x^3 + 4$



22/10/24

Output

Clear

```
Best State Found: [0.016518676849809344, -0.009458567814937036, -0.079479314922839, -0
    .034583502919782516, 0.016824134456549622]
Best Cost Found: 0.008158362864996674

=== Code Execution Successful ===
```