Lab-05

Simulated Annealing Algorithm

Step1: Initialization

1. Choose initial solution 'S'.
2. Set initial temperature T
3. Set a stopping criteria [target achieved]

Step2: Iteration

while (stopping criteria not met)

- Generate neighbours: create neighbouring solution 'S' from 'S'
- Calculate energy difference

Compute cost of E(S) and neighbouring E(S')

IF ($F(S') < (E(S))$) → accept 'S'

else

accept S with probability

$$P = \exp\left\{\frac{E(S) - E(S')}{T}\right\}$$

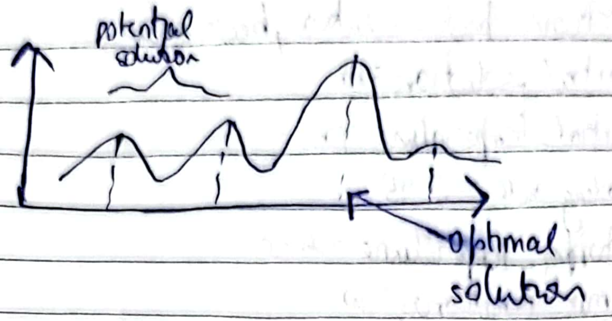
update S to S' as new solution

Return the temperature:

$$T = T - 0.015$$

CS CamScanner

Termination : Return the best solution



Code:

```
import math
import random

def simulated_annealing (objective_function, initial_solution,
                                initial temperature , cooling_rate,
stopping _temperature ,max_temperature)
                  current-solution = initial_solution

                best:solution = current-solution
               best_value = current-value

               temperature = initial_temperature
                iterature =0

          current-solution = new_solution
                 current_value = new_value
           if current_value < best_value:
                best_solution = current-solution
                 best-value = current-value
          temperature = temperature * cooling-rate
          iteration +=1
          print (f" iteration : iteration }
```