CS 312: Task 3

Akshat Karani (170010003, Ojas Raundale (170010004)

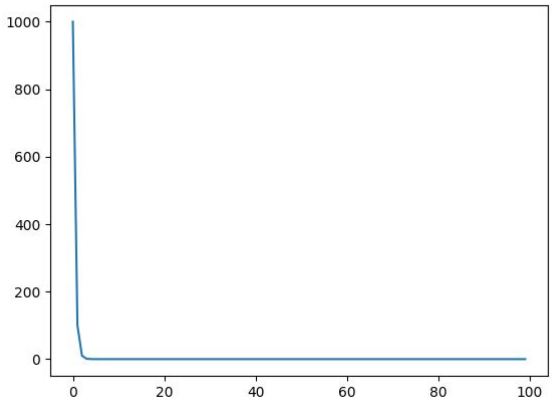
1. Simulated Annealing Analysis and Observation

a. Perturbation method chosen to generate a neighbour

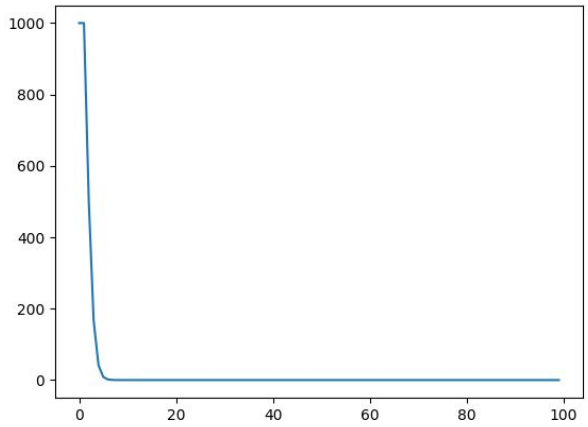
We used two city exchange method to generate a random neighbour. We exchanged two random cities. We chose this method because it is the most intuitive way to generate a neighbour and will generate a different neighbour each time.

b. Cooling schedules tried

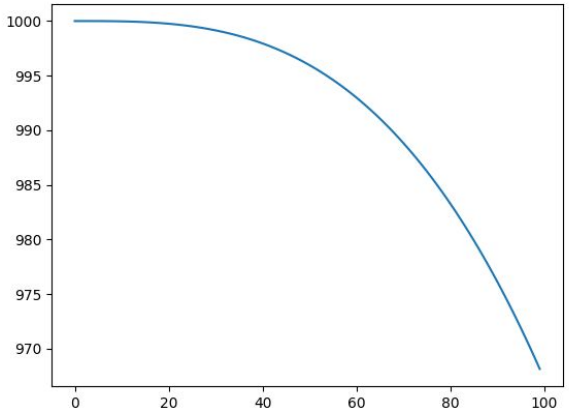
1. temperature = temperature / 10



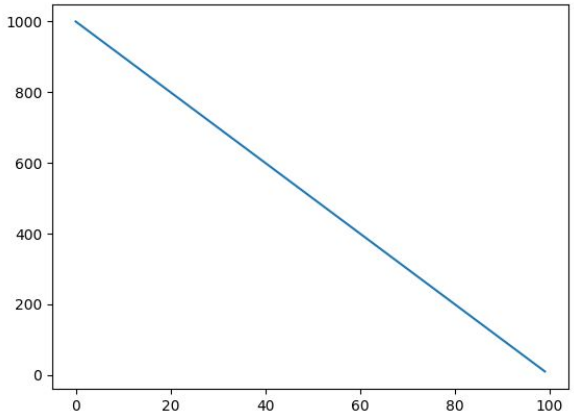
2. temperature = temperature / (time + 1)



3. temperature = temperature - ((time / num_of_epochs) ** 2)



4. temperature = temperature - 10



Initial value of temperature is in thousands, num_of_epochs is the maximum number of iterations which is in hundreds and time is the current iteration. Above graphs show how temperature decreases for 100 iterations.

c. Results

Input Graph	Cooling Schedule	Cost
euc_100	1	2929.9273818574597
	2	2568.110277822511
	3	8722.076013252325
	4	8486.621794685074

euc_250	1	11749.637977945467
	2	8049.833896497353
	3	24673.74125408897
	4	24226.09889266041
noneuc_100	1	5795.3243211325
	2	5979.733200648999
	3	9302.571286271905
	4	9152.729487037703

d. Effect of cooling schedule on tour found

We tried many different cooling schedules in which temperature decreases in different fashion and comparison of results can be seen above. The best result was seen when temperature decreases rapidly first and then slowly.

2. Genetic Algorithm Analysis and Observation

a. Representation chosen

We used Path representation because it is the most intuitive and easy to implement.

b. Crossover Operator

We used two crossover operators, order crossover and cycle crossover. Both the crossover operators are easy to implement.

c. Results

These are result when population size = 50

Input Graph	Crossover Operator	Cost
euc_100	Order	2847.40549380407
	Cycle	4024.4011593708406
euc_250	Order	7420.433636612496
	Cycle	11967.521138146569

noneuc_100	Order	5850.010136924898
	Cycle	6380.440180467398

d. Effect of population size

Result for population size = 500 using Order crossover

Input Graph	Cost
euc_100	1893.16414528575
euc_250	7295.3783690021555
noneuc_100	5669.687201967499

As population size increased cost did not decrease significantly. But the downside of this is that computation time also increased.

3. Ant Colony Optimisation Analysis and Observation

a. Results and effect of number of ants

Input Graph	Number of Ants	Cost
euc_100	20	1833.6542808498093
	100	3286.12862759153
euc_250	20	2827.60402814472
	100	3250.53689558742
noneuc_100	20	5323.055123065402
	100	9589.665831783597

As the number of ants increased, the cost of the tour increased. This is probably because other constants (like alpha, beta, initial value of pheromone) were optimized for the number of ants = 20.
This also increases the cost of computation significantly.

4. Conclusion

a. Comparison

Input Graph	Algorithm	Best Cost
euc_100	Simulated Annealing	2568.110277822511
	Genetic Algorithm	2847.40549380407
	Ant Colony	1833.6542808498093
euc_250	Simulated Annealing	8049.833896497353
	Genetic Algorithm	7420.433636612496
	Ant Colony	2827.60402814472
noneuc_100	Simulated Annealing	5795.3243211325
	Genetic Algorithm	5850.010136924898
	Ant Colony	5323.055123065402

b. Conclusion

From the above results we can see that Ant Colony Optimization is by far the better of three algorithms. Moreover, Ant Colony took the least amount of computation time.
Results given by Simulated Annealing depend majorly on the cooling function used. Although computation time is much more there might be some cooling function which can give better results.
Results given by Genetic Algorithm depend majorly on crossover function used. Mutation can help in getting some unexpected tours which can be good or bad.