CS312: Artificial Intelligence Laboratory
Lab 3 - Stochastic Search

Soumya Srividhya, 170010038
Anudeep Tubati, 170010039

OVERVIEW

This report elaborates on the details of the algorithm and results obtained for the Stochastic searches, namely, Simulated Annealing, Genetic Algorithm and Ant Colony Optimisation. The interpretation of the results and conclusion have also been reported.

I. SIMULATED ANNEALING

Perturbation Method

As our solution vector contained the vertices in the order they are visited, we chose to use the random 2-City swap method to perturb our solution. This also makes our algorithm run faster (other perturbation methods would require one extra for loop as compared to this one) and less convoluted.

Cooling Schedules

Start with Temperature 100 (For euc_100)

1. Linear Cooling Function (Slope = -0.01)

   Best Tour Found = 44 90 42 67 40 3 83 7 21 92 26 16 72 33 55 66 8 73 75 20 41 98 61 58 37 78 89 35 71 77 87 43 79 50 70 62 45 56 23 82 68 94 53 91 0 74 18 36 63 48 97 4 19 76 69 14 39 5 47 6 51 31 49 95 24 81 34 60 38 65 80 99 15 29 54 11 22 1 9 30 32 46 93 64 25 12 88 2 59 86 96 27 28 57 10 85 52 13 84 17

   Best Tour Length = 3815.3

2. Linear Cooling Function (Slope = -0.1)

   Best Tour Found = 9 17 59 99 67 40 90 61 50 37 71 15 85 11 72 27 60 34 92 47 38 53 23 49 45 86 33 55 77 94 14 0 56 68 19 4 22 26 16 10 75 25 41 18 13 52 24 76 39 5 3 89 62

65 81 21 51 91 82 7 6 95 31 83 69 78 2 73 8 64 84 80 42 87 43 35 58 44 96 74 63 1 66 93 12 32 54 98 36 20 88 29 57 28 97 48 46 79 70 30

Best Tour Length = 4681.24

3. Hyperbolic Cooling Function (Iters = 10000)

   Best Tour Found = 54 48 63 1 41 36 2 73 8 66 87 58 45 65 38 0 85 29 98 93 61 50 39 76 24 47 6 27 28 97 88 20 32 25 79 35 3 82 51 91 7 80 84 33 17 77 59 56 49 31 23 53 90 40 94 62 71 9 70 44 99 86 13 52 15 43 37 67 68 69 5 34 81 72 22 74 60 19 4 11 16 26 57 10 18 64 55 30 46 75 12 42 89 78 83 95 96 92 21 14

   Best Tour Length = 4082

Results

Linear Cooling Function (Slope = -0.01)

	euc_100	**euc_250**	**noneuc_100**
Best Length	4100.93	10029.4	6010.51

Effect of Cooling Schedule on Tour Found

The cooling schedules which cool down slowly seem to work better than the ones which cool down fast. This is because the cooling schedule which cools down fast induces more of a Hill Climbing nature into the algorithm and hence the algorithm has a higher chance of getting stuck at a local maxima.

II. GENETIC ALGORITHM

Representation

Path Representation method is chosen to represent a Tour as it is the most natural way to represent the Tour (i.e., vertices in the order in which they were visited). Also, compared to Adjacency crossover, the Path representation method had much better crossover operators available. The Ordinal representation took more time and space to be maintained, as extra work on Path representation has to be done to get it’s Ordinal representation.

Crossovers

We explored three crossover operators using path representation. The three operators are Partially Mapped Crossover (PMX), Cycle Crossover and Order Crossover. Among them we chose the PMX operator, as it gave the best results.

Cyclic crossover tries to retain the maximum tour from its parents, hence seldom explores new tours. The Order crossover is the fastest among the 3, but doesn't give as good results as the PMX crossover. This is because Order crossover is more random compared to PMX, as it selects a subtour and places all the leftover vertices into the tour continuously, hence alters the parent too much. Whereas in PMX, it systematically fills in the child's tour after selecting a subtour.

Results

Order Crossover Operator:

Population Size	Graph		
	euc_100	euc_250	noneuc_100
100	9107.74	23796.7	8920.75
1000	8806.26	23225.3	8857.51
5000	8695.57	23548.6	8857.51
10000	8596.24	23462.8	8971.69

Cycle Crossover Operator:

Population Size	Graph		
	euc_100	euc_250	noneuc_100
100	9534.87	24853.6	9483.94
1000	9464.34	24077.4	9336.76

5000	9341.59	23755.5	9271.16
10000	9236.5	23333.8	9229.37

Partially Mapped Crossover Operator:

Population Size	Graph		
	euc_100	**euc_250**	**noneuc_100**
100	8940.44	24612.3	9218.03
1000	8874.66	24012.1	9040.14
5000	8763.54	23309.4	8918.31
10000	8674.1	23197.8	8837.43

Effect of Population Size on Tour Found

As the population size increases the length of the Tour found is decreasing in almost all cases. Hence, we could say population size is inversely proportional to the length of the Tour found. The increase in the population increases the variety of the population, as even less fitter parents get a chance to clone and the probability of choosing better parents also goes up, which results in better evolution of the offspring.

III. ANT COLONY OPTIMISATION

Results

Graph	Best Length		
	Ants = 100	**Ants = 500**	**Ants = 1000**

euc_100	9301	9122	9038
euc_250	24378	24277	23644
noneuc_100	9277	8957	8988

Effect of Number of Ants on Tour Found

As more ants will explore better and have a greater chance to find a shorter path (which will then be followed by the other ants because of pheromones), more ants tend to give a better solution.

CONCLUSION

Out of the 3 algorithms implemented, Simulated Annealing gave us the best results. It proved to be the fastest one to execute as well. It doesn't need much information about the system, hence can deal with arbitrary systems and cost functions. However, the schedule might be tough to figure out (also slow to calculate), as not every problem can be solved with the same schedule.

Genetic Algorithm seems to require more information about the problem, as designing a suitable representation and a crossover operator might prove to be tedious. It gave decent results in the experiment and guarantees to give a better solution with increasing population (as there's more variety). It is time-consuming as there are a lot of parents and each pair of them have to be crossed over.

Ant Colony Optimisation gave worse results than Genetic Algorithm. It is also computing-heavy as it needs to simulate a number of ants (who in turn have an N-city tour) every iteration. It also doesn't take into consideration the pheromone dropped when each ant travels an edge, as it only updates the pheromone after the whole tour is completed. However, this algorithm can be much more efficient than the others given a parallel execution framework. It also maintains a good intuition of more pheromone and less distance.

Conforming to the observations, we'd choose Simulated Annealing to solve the TSP.