CS312: Artificial Intelligence Laboratory TSP Competition Report

Soumya Srividhya, 170010038
Anudeep Tubati, 170010039

OVERVIEW

This report elaborates on the methodology used for solving the Travelling Salesman Problem and the improvements done vis-à-vis the previous lab.

I. METHODOLOGY

We used Ant Colony Optimization algorithm for this round. It was found that ACO was doing much better (as compared to the other two algorithms) when coded in Python rather than C++

II. IMPROVEMENTS

As there are a couple of hyperparameters to tune in ACO (like alpha, beta, rho etc.), we searched the hyperparameter space (by testing some values which we thought were intuitive) to find the set of values which gave the best results.

Apart from this, we observed that the algorithm performs better when the population of ants is proportional to the number of cities. We also tried a variation of ACO, in which we update the pheromone for only the top ants (top in the sense of shortest tours found). This modification proved to work really well as we observed a significant improvement in the performance and result of the algorithm.

Values we achieved

	euc_100	**euc_250**	**noneuc_100**
ACO	1565.37	2631.84	5386.72

III. CONCLUSION

We chose ACO (Python) as it was giving the best results among all the three, as compared to our last round in which we had used Simulated Annealing (C++). ACO also has better performance compared to the other two algorithms.

Keeping ants proportional to the cities is optimal as higher ants give better results but also perform slower. In a general sense, it is better to keep the population of ants proportional to the size of the graph.

Updating only the top ants works better as the tours with higher costs are neglected and this results in more ants traversing on shorter paths.