CS6380 - Artificial Intelligence

Assignment 1 - Travelling Salesman Problem

CS11B028 - Viswajith V
CS11B058 - Sriram V

1 Approach Used

We used some constructive heuristics to come up with initial guesses, and then perturbed these solutions while simultaneously running a genetic algorithm through the current solution population.

2 Constructive Heuristics

We used three different kinds of constructive heuristics for our starting solutions; each of them give decent approximation for random problem instances. There are specific types of problems where each heuristic is weak; it is to neutralise this effect that we used all of them.

2.1 Minimum Spanning Tree-based algorithm

This algorithm constructs a tour whose cost is within two times the optimal tour cost (ie, a 2-approximation) for Euclidean problem instances. It computes the MST for the given distance graph (we have used Prim's algorithm to accomplish this) and, starting from an arbitrary node, reports the nodes as they are visited in a pre-order (depth-first) traversal.

2.2 Nearest Neighbour Tours

This is a greedy heuristic; pick a random start city, and from each stage pick the closest unvisited node. It generates different tours for different start point cities; we generate a handful of such tours.

2.3 Cheapest Link Tour

This algorithm is based on choosing the cheapest possible edges at each stage, while ensuring that at no point does any vertex have more than two edges connected to it. It is starting point-agnostic, and thus produces only one tour.

3 Perturbing the initial tours

3.1 2-opt

We make each constructed tour two-optimal, ie, we consider all possible pairs of edges, and if exchanging them would reduce the tour cost, we exchange them. We do this until there is no such pair of edges. At this stage, the tour is two-optimal.

3.2 3-opt

We run this on the two-optimal tours to make them three-optimal; here, we consider all possible combinations of 3 edges, and consider the various ways of exchanging them; if one or more of the possible exchanges improves our tour, we perform the best exchange. We do this until no more such exchanges are possible which would improve our tour. At this stage, the tour is three-optimal.

4 Genetic Algorithm

We initialise a population of candidate tours using the tours generated earlier and some random starting tours. We then evolve our population over one generation. We create a tournament by randomly picking a subset of the population, and pick the parent from among them; this automatically gives a higher probability of selection for fitter parents. We then do a partially mapped crossover (PMX) to generate the children. We retain a certain fraction of the parents, and fill the rest up with children. We mutate some children with a fixed mutation rate; the mutation is a simple two-edge exchange.

Now, from the next generation, we pick the fittest few, and also the fittest few from among the newly generated children. We make these tours two- and three- optimal, and then use them to initialise another evolution using our genetic infrastructure.