CSE 4633 : Intro to Artificial Intelligence

Naveen Neelakandan (nn149)

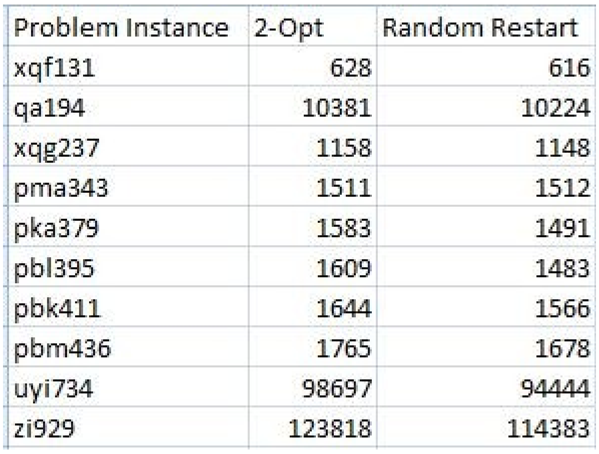
The traveling salesman is a combinatorial optimization problem which is NP-hard in nature. It involves finding the shortest Hamiltonian cycle in a graph. We model the problem as a undirected, weighted and complete graph. It is further assumed that the problem is symmetric in that the distance between any two cities is the same in any direction. This is the model we assume in solving the problem. The distance between 2 cities is represented in a distance matrix where matrix[x][y] represents the distance between cities x and y.

As the traveling salesman is an NP-hard problem, systematic search algorithms for it take exponential time in the worst case. Local search, an alternative finds near-optimal solutions much faster than systematic search. However, local search is not guaranteed to find an optimal solution to the problem. In this assignment, we utilized 2-opt a well -known local search algorithm on the traveling salesman. The performance of 2-opt is then improved by supplementing it with random restarts.

The 2-opt local search algorithm takes an initial tour and randomly selects two edges to remove fro m the tour. The cities are then reconnected by 2 new edges. It is noted that there is only one valid way to reconnect the cities. If we let s1 and d1 be the cities connected by the first edge and s2 and d2 be the cities connected by the second edge, then the only way to connect these cities differently is to add edges from s1 to s2 and d1 to d2. The tour d1 to s2 is inverted in the process so that we visit s2 first. The algorithm terminates when there is no improvement in the tour length for a specified number of iterations. From the implementation of the algorithm, it is seen that it runs in minimal time. Naturally, as the number of cities in the tour increases ,the time taken to compute the length of the solution also increases. The time taken to find a solution is also noted to not rest on the distance of the tour as this does adds little to the complexity of the calculations. Furthermore, it is found that for best performance (most near optimal solution), the specified number of iterations to terminate after should be increased as the number of cities is increased. The 2-opt does not however produce optimal solution no matter how long it runs. This is because the algorithm is stuck in local maxima and in order to get to the optimal solution, it may be necessary to accept a tour of longer length, which the algorithm prohibits. The optimality of the algorithm though is also found to depend on the original starting state. Some starting states routinely give better solutions while others do not. This leads us to the next algorithm, the two-opt with random restarts.

Random restarts help us to avoid the problem of getting trapped in local optima to a certain extent as by restarting the algorithm with different starting states, we may not get trapped in the same local optima leading to better solutions. In implementing the algorithm, it is found that there is not much difference in solution quality as compared to the 2-opt for a small number of cities. However, as the number of cities rises, the solution quality of 2-opt with random restarts is found to be much better than just the 2-opt. A higher number of restarts also naturally leads to better solutions as there are more random starting states to run the algorithm from.

Therefore, in implementing 2-opt and the 2-opt with random restarts it is seen that local search can be a valid alternative to systematic search for cases in which we are more concerned about the time taken to find the solution as compared to the solution optimality. The optimality of local search can be further improved by supplementing it with techniques such as simulated annealing and random restarts.



Problem Instance	2-Opt	Random Restart
xqf131	628	616
qa194	10381	10224
xqg237	1158	1148
pma343	1511	1512
pka379	1583	1491
pbl395	1609	1483
pbk411	1644	1566
pbm436	1765	1678
uyi734	98697	94444
zi929	123818	114383

Figure 1: Comparison of 2-Opt by itself and 2-Opt with Random restarts

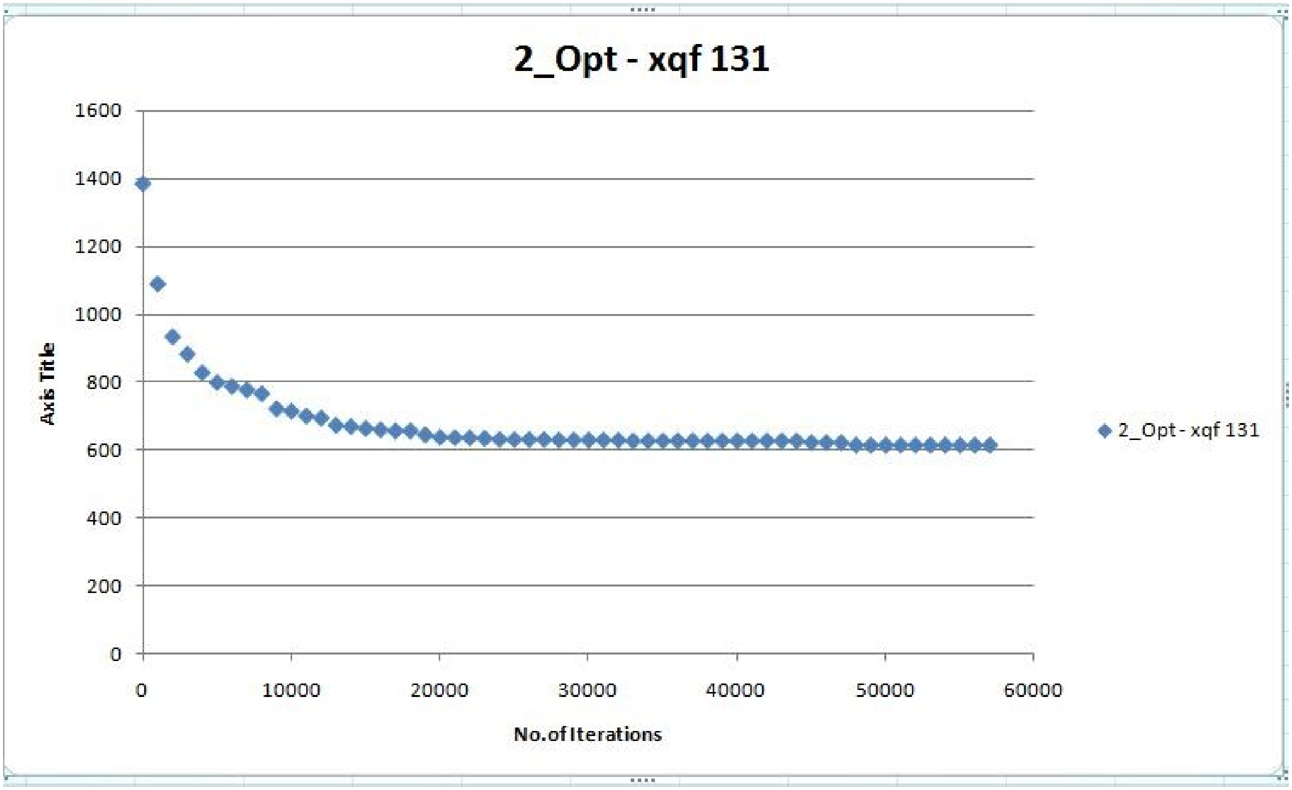


Figure 2: Graph of quality of the tour as a function of number of iterations