

Assignment no. 10

Title: Branch and Bound

Problem:

Suppose you are given a set of n cities and a distance matrix that gives the distance between each pair of cities. Your task is to find the shortest possible tour that visits every city exactly once and returns to the starting city.

Solution using a Branch and Bound Algorithm: One approach to solve this problem is to use the branch and bound algorithm. The algorithm works as follows:

1. Begin with a starting city and an empty tour.
2. Select a city that has not yet been visited and add it to the tour.
3. If all cities have been visited, calculate the total distance of the tour and update the current best solution if the current tour has a lower total distance.
4. If the total distance of the current tour is already greater than the current best solution, backtrack and remove the last added city.
5. If there are remaining cities, branch into subproblems for each unvisited city: one in which the next city is the closest unvisited city, and one in which the next city is the second closest unvisited city.
6. For each subproblem, calculate a lower bound on the total distance by assuming that the remaining cities are visited in the closest order, and add it to a priority queue based on the lower bound.
7. Select the subproblem with the highest priority (i.e., the smallest lower bound) and repeat from step 2.

The key idea of this algorithm is to use the lower bounds to eliminate subproblems that cannot lead to a better solution than the current best solution, and focus on the subproblems with the highest potential for improvement.

Tasks:

1. Analyze the complexity of this solution
2. Can you make the solution better by tweaking it here and there?