

Travelling Salesperson Problem using Branch and Bound algorithm

The Branch and Bound algorithm for the Travelling Salesperson Problem can be summarized in the following steps:

1. Start with an initial upper bound on the cost of the optimal tour, which can be obtained using a heuristic or by solving a relaxed version of the problem. Set the current best solution to be the upper bound.
2. Create an initial node representing the problem of finding a tour that starts at a given city (say, city 1)
3. Compute the lower bound on the cost of the tour represented by the node. This can be done using a lower bound heuristic, such as the minimum spanning tree (MST) or nearest neighbor (NN) algorithm.
4. If the lower bound on the cost of the tour represented by the node is greater than or equal to the current best solution, prune the node and backtrack to the previous node.
5. If the lower bound on the cost of the tour represented by the node is less than the current best solution, create child nodes representing the sub-problems of finding tours that start at each of the unvisited cities, and compute their lower bounds.
6. Sort the child nodes in increasing order of their lower bounds, and explore them one by one, starting with the node with the lowest lower bound.
7. For each child node, repeat steps 4 to 6 until all nodes have been explored or pruned.
8. If all nodes have been explored and no better solution has been found, the current best solution is the optimal solution. Otherwise, backtrack to the previous node and continue exploring the search tree.

Repeat steps 2 to 8 until all possible tours have been explored.

The Branch and Bound algorithm for TSP uses a depth-first search approach to explore the search tree, pruning sub-trees that cannot contain the optimal solution. By using lower bounds to guide the search, the algorithm can often find an optimal solution faster than a brute-force approach. However,

the time complexity of the algorithm can still be exponential in the worst case, making it impractical for very large TSP instances.

The Travelling Salesperson Problem (TSP) is an NP-hard optimization problem that seeks to find the shortest possible route that visits a set of cities exactly once and returns to the starting city. The Branch and Bound algorithm is one of the most commonly used techniques for solving TSP.

The basic idea of the Branch and Bound algorithm is to divide the problem into smaller sub-problems and solve each sub-problem recursively. The algorithm starts with an initial upper bound, which is updated as it explores the solution space. At each step, the algorithm branches out to explore new solutions, but only continues to explore a branch if the lower bound on its cost is lower than the current upper bound.

The time complexity of the Branch and Bound algorithm for TSP depends on the size of the problem and the quality of the initial upper bound. In the worst case, the algorithm has an exponential time complexity of $O(n!)$, where n is the number of cities. However, in practice, the algorithm can be significantly faster than brute force methods, especially for large problems, due to its ability to prune the search space using lower bounds.

The time complexity of the Branch and Bound algorithm can be improved by using heuristics to obtain better initial upper bounds and lower bounds. Additionally, parallelization techniques can be used to explore different branches of the search tree simultaneously, further reducing the time required to find an optimal solution.