Branch and bound algorithm for TSP

1) Get upper bound (e.g. by greedy algorithm).

2) Solve the problem as an assignment problem (without constraint that tour be connected), e.g. using TORA or Hungarian method. Record cost and cycle structure of the solution. This forms the root of a tree.

3) (BOUNDING, I) Each time an assignment problem has been solved and gives a cost worse than the current upper bound, terminate that branch.

4) (BRANCHING) If a solved problem doesn't give a valid tour, pick one cycle in the tour (usually the shortest), and create branches, one for each edge in the cycle. At the end of each branch, solve the assignment problem corresponding to the parent problem together with the additional constraint that the associated edge is forbidden from the solution. (To forbid an edge, set its cost to ∞).

5) (BOUNDING PART II) If a solved problem yields a valid tour, with cost ($z^\star$, say) that is less than the current upper bound, then replace the current upper bound with $z^\star$ then terminate all branches whose objective value is worse than $z^\star$.

6) Stop when only one branch survives. This solves the TSP.