

Approximate Traveling Salesperson (TSP) Tour Construction (Doubling MST)

This is a handout for another version of the Approximate TSP Tour Construction Algorithm given on page 119 of the textbook (4th ed.). Given a TSP instance on a graph G with an associated cost matrix C , the algorithm goes through three main steps:

1. Find a minimal spanning tree (MST) on G by using Prim's or Kruskal's algorithm. Let T be the resulting MST.
2. (Doubling) For each edge $(i, j) \in T$, add another edge between i and j with the same cost c_{ij} . Note that the subgraph consisting only of the edges in T and these duplicate edges has an Euler cycle (why?).
3. (Rounding) Pick any vertex i . Find an Euler cycle P that starts and ends at i . Let $P = (x_1, x_2, \dots, x_n)$, where $x_1 = x_n = i$. Trace the Euler cycle P and delete the repeated vertices until you are left with a TSP tour.

Example: Consider the TSP instance whose cost matrix C is given by

	1	2	3	4
1	∞	3	9	7
2	3	∞	6	5
3	9	6	∞	6
4	7	5	6	∞

This is a TSP instance on K_4 . Using Kruskal's or Prim's algorithm, we find that one MST is given by $T = \{(1, 2), (2, 3), (2, 4)\}$. Double each edge in T . Now an Euler cycle in the resulting subgraph is given by $P = (1, 2, 3, 2, 4, 2, 1)$. So, we start tracing the vertices in P and remove the repeated vertices to get $(1, 2, 3, 4, 1)$, which is a TSP tour.

Theorem 1 *If the edge costs are nonnegative and symmetric ($c_{ij} = c_{ji}$ for all vertices i, j) and they satisfy the triangle inequality (i.e., for any triple of vertices i, j, k , we have $c_{ik} \leq c_{ij} + c_{jk}$), then the cost of the approximate TSP tour is no more than twice the cost of the minimal TSP tour.*

Proof: Let T be an MST on graph G . The cost of T is less than or equal to the minimal TSP tour. Because, if you remove one edge from the optimal TSP tour, you get a spanning tree. But T is a minimal spanning tree and the edge costs are nonnegative. Double the MST and let P be an Euler cycle on the resulting subgraph. Clearly, the total cost of the Euler cycle is twice the cost of MST. Note that step 3 of the algorithm obtains a TSP tour as follows: If P has a sequence like i, l, \dots, j, i, k , then we replace it by i, l, \dots, j, k (i.e., we delete the second (repeated) i). Note that the only difference in the total cost of these two sequences arises from the deletion of the second i . We have a common cost term plus $c_{ji} + c_{ik}$ in the first one as opposed to the same common cost term plus c_{jk} for the second sequence. But by the triangle inequality, $c_{jk} \leq c_{ji} + c_{ik}$. Therefore, rounding step does not increase the cost of the Euler cycle. Consequently, the total cost of the TSP tour obtained this way is no more than twice the cost of the Euler cycle, which in turn is no more than twice the cost of the minimal TSP tour. QED

Remarks:

1. The cost of the optimal TSP tour for the given example is 22 (there are two such tours: 1-2-3-4-1 and 1-4-3-2-1 (by symmetry)). Therefore, the cost of any TSP tour obtained this way will not be more than 44. In particular, the approximate TSP tour we constructed turns out to be optimal. This, however, is not true in general.
2. The performance of the algorithm depends on the Euler cycle P . If instead, we had picked $P_a = (1, 2, 4, 2, 3, 2, 1)$, then our approximate TSP tour would be $(1, 2, 4, 3, 1)$ with a cost of 23.
3. The performance of the algorithm also depends on the choice of the starting vertex. For instance, if we build an Euler cycle starting at vertex 2, then $P_b = (2, 3, 2, 1, 2, 4, 2)$, which would yield the TSP tour $(2, 3, 1, 4, 2)$ with a cost of 27.
4. The best strategy is to run the algorithm several times using different Euler cycles and different starting vertices and pick the tour with the lowest cost.