P problems used in approximating NP-hard problems

A minimum spanning tree is the set of edges that connect the vertices of an undirected weighted graph without creating cycles and using the minimum sum of edge weights. According to Wigderson (2009), the minimum spanning tree problem is an example of a P problem because a solution can be efficiently computed. The two greedy algorithms used to find minimum spanning trees are Prim's algorithm and Kruskal's algorithm. Minimum spanning trees are used to approximate an NP-hard problem: the traveling salesman problem.

Prim's algorithm solves the minimum spanning tree problem by starting with an empty tree, picking any vertex of the graph to be the first element in the tree then picking the minimum edge that connects a tree vertex to a non-tree vertex and repeating this until all vertices in the graph are part of the tree.

Kruskal's algorithm on a graph with V vertices begins by initializing V separate sets each with one vertex, considering edges from smallest to largest and when an edge joins two sets, using it to join the sets and returning the eventual combined set of vertices.

The traveling salesman problem asks to find the shortest route such that a salesman visits a given number of cities and gets back to the original city. Getting back to the original city implies that there would be a cycle in the graph. Since minimum spanning trees have no cycles and have the minimum possible sum of edge-weights, the sum of a minimum spanning tree's edge weights is less than that of the traveling salesman. The sum of a minimum spanning tree's edge weights is also shown to be at most twice that of the traveling salesman, this is because if the one followed the path through the minimum spanning tree and reversed along the same path, they would have gone through the edge weights twice and got back to the original vertex.

Therefore, the sum of edge weights in the traveling salesman problem is greater than that in a minimum spanning tree but less than that of twice the minimum spanning tree.

References

Stewart, I. (2000). *Million Dollar Minesweeper*. Retrieved from http://www.minesweeper.info/articles/MillionDollarMinesweeper.pdf

Wigderson, A. (2009). *Knowledge, Creativity and P versus NP*. Retrieved from http://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/AW09/AW09.pdf

Vazirani, U. *Dynamic Programming* [PDF document]. Retrieved from Lecture Notes Online Website: https://people.eecs.berkeley.edu/~vazirani/algorithms/chap6.pdf

Tov, J. *Minimum Spanning Tree* [PDF document]. Retrieved from Lecture Notes Online Website: http://users.eecs.northwestern.edu/~jesse/course/eecs214-fa17/lec/14-mst.pdf