Approximation Algorithm for TSP with Triangular Inequality

Restrictions on the weighted, undirected graph $G=(V, E)$:

1. There is an edge connecting every two distinct vertices.
2. Triangular Inequality: If $W(u, v)$ denotes the weight on the edge connecting vertex $u$ to vertex $v$, then for every other vertex $y$,

\[ W(u, v) \leq W(u, y) + W(y, v). \]

NOTES:
- These conditions satisfy automatically by a lot of natural graph problems, e.g., cities on a planar map with weights being as-the-crow-flies (Euclidean distances).
- Even with these restrictions, the problem is still NP-hard.

A simple TSP approximation algorithm:

Step 1. Determine a Minimum Spanning Tree (MST) for $G$ (e.g., Prim's Algorithm section 4.1)

Step 2. Construct a path that visits every node by performing a preorder walk of the MST. (A preorder walk lists a tree node every time the node is encountered including when it is first visited and "backtracked" through.)

Step 3. Create a tour by removing vertices from the path in step 2 by taking shortcuts.

1) (Step 1) Determine a Minimum Spanning Tree (MST) for $G$ (e.g., Prim's Algorithm) if we start with vertex 1 in the MST. (Assume edges connecting all vertices with their Euclidean distances)

Prim's algorithm is a greedy algorithm that performs the following:

a) Select a vertex at random to be in the MST.
b) Until all the vertices are in the MST:
   - Find the closest vertex not in the MST, i.e., vertex closest to any vertex in the MST
   - Add this vertex using this edge to the MST

2) (Step 2) Determine the preorder walk of the MST.

\[ 1, 2, 3, 8, 3, 2, 6, 5, 7, 5, 4, 5, 6, 2, 1 \]
Thus, your obtained from the preorder-walk of the MST > 2 * the optimal TSP tour

0.5 * tour obtained from the preorder-walk of the MST > the optimal TSP tour

Where is the relationship between the tour obtained from the preorder-walk of the MST and the optimal TSP tour?

0.5 * tour obtained from the preorder-walk of the MST >> distance of the preorder-walk of the MST

Furthermore, when in the preorder-walk, then: distance of the MST > 2 * distance of the preorder-walk of the MST. Because every edge in the MST is

distance of the MST -> spanning tree by removing edge from optimal tour > the optimal TSP tour

A spanning tree since it includes all the vertices and contains no cycles.

(b) When scanning the above path, how did you know which vertices to eliminate to take a shortcut?

(c) Given removing vertices from the preorder-walk path to create a tour by

TSP tour

1. Complete a tour by removing vertices from the path in step 2 by taking shortcuts.
Complete the backtracking search tree with pruning.

1. If the partial solution is a dead end, then... (insert explanation)
2. If the partial solution is better than the current best solution...
3. If the partial solution is not a dead end...

Backtracking General Idea: (Recall the coin-change problem from Lecture 10 and 13)

Data Structures (CS 1520)
b) Complete the best-first search with branch-and-bound state-space tree with pruning. Indicate the order of nodes expanded.

(a) Which type of data structure would we use to find the most promising node to expand next? Why?
- Expands the most promising ("best") node first by visiting its children
- "Promising" node might be calculated a "bound" estimate for each node that could be obtained from any node in the subtree rooted at that node, e.g., how "promising" a node might be
- Does not limit us to any particular search pattern in the state-space tree

3. In the best-first search with branch-and-bound approach:
   - Name:  
   - Lecture 29