Combinatorial Optimization – Problem Set 11 – Solutions

Metric TSP

1. *Give tight examples for the Double-Tree algorithm and Christofides' algorithm (i.e. give examples which show that these are not k-approximation algorithms for any $k < 2$ and $k < 3/2$, respectively).*

   Double-Tree: In a complete graph G on n vertices, consider a subgraph H consisting of an $n-1$-cycle, and the remaining vertex with edges to each of the $n-1$ vertices on the cycle (so H is a union of a cycle and a star). Give the edges of H weight 1, and give all other edges of G weight 2.
   There is a Hamilton cycle in H of weight n, which is clearly minimum. But the Double-Tree algorithm may end up with a Hamilton cycle of weight $2n-2$, which proves the statement.
   Indeed, one of the MSTs of G is the star from H. If the Euler tour goes through the doubled star in the wrong way, not visiting after another any of the vertices that are adjacent on the cycle (you might want to draw a picture here), then shortcutting will give a Hamilton cycle that uses $n-2$ edges of weight 2, and 2 of weight 1.

   Christofides: We will force the algorithm to pick a bad edge, by designing the graph so that there is an MST which is a path whose endpoints are the vertices of that bad edge. Then the bad edge is the only matching of the odd vertices on the path, so Christofides will add the bad edge. At the same time we make sure there is a shorter Hamilton cycle which avoids the bad edge. Actually, for the construction it is a bit simpler to start with the cycle and then add the path.
   Start with a cycle $C = v_1 v_2 \cdots v_n v_1$ on n vertices. Now add the following path P of length $n-1$ from v_1 to its "antipode" $v_{\lfloor n/2 \rfloor}$:

   $$P = v_1 v_2 v_n v_3 v_{n-1} v_4 v_{n-2} \cdots v_{\lfloor n/2 \rfloor}.$$

   Call this graph H.
   Construct the distance graph G of H: A complete graph with $V(G) = V(H)$, with weights $d(uv) = \mathrm{dist}_H(u, v)$, the length of the shortest path between u and v. This graph automatically satisfies the triangle inequality, and the minimum Hamilton cycle is C, of weight n. Note that $d(v_1 v_{\lfloor n/2 \rfloor}) = \lfloor n/2 \rfloor$, because P creates no shorter path from v_1 to $v_{\lfloor n/2 \rfloor}$ than going around one half of C.
   If the MST used in Christofides is the path P, then the only odd-degree vertices are v_1 and $v_{\lfloor n/2 \rfloor}$. The only possible matching is then $M = \{v_1 v_{\lfloor n/2 \rfloor}\}$. Adding $v_1 v_{\lfloor n/2 \rfloor}$ to P gives a cycle, which will be the resulting Hamilton cycle, with weight $d(P) + d(v_1 v_{\lfloor n/2 \rfloor}) = n - 1 + \lfloor n/2 \rfloor$, which goes to $3n/2$ as $n \to \infty$.

2. *Show that a k-approximation algorithm for Metric TSP gives a k-approximation algorithm for TSPR.*

> Travelling Salesman Problem with Repetitions (TSPR): Given an undirected complete graph G, with weights $w : E(G) \to \mathbb{R}_{\geq 0}$, find a tour (closed walk) that visits every vertex *at least* once.

Let G be a graph with weights w in which to find a tour that visits every vertex at least once. Construct its distance graph G': A complete graph with $V(G') = V(G)$, with weights $w'(uv)$ defined as the weight of the shortest path between u and v. This can be constructed in polynomial time because the weights are nonnegative, by an earlier problem. Clearly G' is metric. We claim that a minimum tour visiting every vertex in G at least once corresponds to a minimum Hamilton cycle in G' of the same weight. This proves the statement.

Let T be a tour in G that visits every vertex. Let $\{v_1, \ldots, v_n\}$ be all the vertices of G, in the order in which they appear in T. Then $H = v_1 \cdots v_n v_1$ is a Hamilton cycle in G'. T can be split into subwalks P_i from v_i to v_{i+1}, and then we have $w'(v_i v_{i+1}) \leq w(P_i)$ for all i by definition of w'. Hence

$$w'(H) = \sum w'(v_i v_{i+1}) \leq \sum w(P_i) = w(T).$$

Note that we do not necessarily have equality here.

Let $H = u_1 u_2 \cdots u_n u_1$ be a Hamilton cycle in G'. Then for every $u_i u_{i+1}$, there is a path Q_i in G from u_i to u_{i+1} of length $w(Q_i) = w'(u_i u_{i+1})$. Let T be the tour obtained by concatenating all Q_i. Then $w(T) = w(H)$.

This proves the claim: A minimum such tour T gives a Hamilton cycle H with $w'(H) \leq w(T)$. If we had $w'(H) < w(T)$, then we could use H to construct such a tour $\widetilde{T}$ with $w(\widetilde{T}) = w'(H) < w(T)$, contradicting minimality of T. So $w'(H) = w(T)$.

3. *Find a 4/3-approximation algorithm for 1-2-TSP.*
(Hint: Cover the vertices by cycles of minimum total weight, then patch them together.)

> Travelling Salesman Problem (1-2-TSP): Given an undirected complete graph G, with weights $w : E(G) \to \mathbb{R}_{\geq 0}$ such that all $w(e) \in \{1, 2\}$, find a Hamilton cycle of minimum weight.

Below we will show how to find a minimum cycle cover. First suppose we have a minimum cycle cover C^*, and let H^* be the minimum Hamilton cycle. Since H^* is also a cycle cover, we have $w(C^*) \leq w(H^*)$. Also $n \leq w(H^*)$ because all weights are ≥ 1. We will repeatedly patch together two cycles two create a new cycle. Since every cycle has at least 3 vertices, we will need at most $\lfloor n/3 \rfloor - 1$ patching steps. We'll show that we can do this in such a way that the total weight of the cycles increases by at most $\lfloor n/3 \rfloor$. Then we will get a Hamilton cycle of weight

$$w(C^*) + \lfloor n/3 \rfloor \leq w(H^*) + n/3 \leq w(H^*) + w(H^*)/3 = \frac{4}{3} w(H^*).$$

That leaves two things to work out: how to patch the cycles while not increasing the weight too much, and how to find a minimum cycle cover.

- *Patching the cycles:* We patch two cycles together by removing one edge from each, say e_1 and e_2, then adding in two new edges (possibly because the graph is complete) to create one large cycle, say e_3 and e_4. The only way that this could increase the weight by more than 1 is if e_1 and e_2 both had weight 1, and e_3 and e_4 both had weight 2; the total weight would then increase by 2. We can avoid this as long as we choose e_1 to have weight 2, which is possible when there is some cycle in the cycle cover that has some edge with weight 2. The only situation where we cannot avoid this is if all edges in the cycle cover have weight 1.
So we might have to increase the weight by 2 in one step. But after that, there will be an edge of weight 2, so we can avoid this from then on. Well, actually, it is possible that we return to the situation where all edges in the cycles have weight 1, but only after a step in which the total weight decreased. So anyway, we can guarantee that after k patching steps, the total weight has increased by at most $k + 1$.
So the $\lfloor n/3 \rfloor - 1$ patching steps will increase the total weight by at most $\lfloor n/3 \rfloor$.

- *Minimum Cycle Cover:* We construct a new graph G'. Replace each vertex v as follows: Put two vertices z_1 and z_2; for each incoming edge e, connect it to a vertex x_e, connect x_e to a vertex y_e, and connect y_e to both z_1 and z_2. The incoming edge has the same weight as in G, every new edge has weight 0. So, for a vertex with k incoming edges, there are k vertices x_e, k vertices y_e, and 2 vertices z_1, z_2.
Find a minimum weight perfect matching for G'. We claim that this corresponds to a minimum weight cycle cover. The weight is clearly the same. For each vertex v, the z_1 and z_2 are matched to some y_{e_1} and y_{e_2}. The other y_e must be matched to the corresponding x_e, so x_{e_1} and x_{e_2} must be matched by the edge with nonzero weight. Viewing this whole thing as one vertex, we see that the vertex is matched by exactly two edges. Hence this is a cycle cover (aka 2-matching or 2-factor).

The cycle cover part seems overly complicated, please let me know if you find something simpler. I should have asked the question for directed TSP, where this part is much easier...