Assignment 5

Yang David Zhou, ID 260517397

November 20, 2014

Problem 1. *Port Authority.*

Solution. I assume that no $w_i > K$.

(a) Let $K = 4$ and $w = \{2, 3, 2, 4, 1\}$. Greedy uses 5 trucks when $OPT = 3$.

(b) Let $W = \sum_{i=1}^{n} w_i$ and the greedy solution be m. When considering any container w_j, the truck being loaded is sent away only if the sum of the container weights on the truck and w_j is $> K$. Every consecutive pair of trucks had total weight between the two trucks $> K$ because the first truck in the pair was sent away. If there are an even number of trucks then $W > \frac{m}{2} \cdot K$. So,

$$\begin{aligned} \frac{W}{K} &> \frac{m}{2} \\ m^* \geq \frac{W}{K} &> \frac{m}{2} \\ m^* &> \frac{m}{2} \\ m &< 2m^* \end{aligned}$$

And if there are an odd number of trucks, say $m = 2q + r$ and set $r = 1$. Similar to the even case, we have $W > q \cdot K$. So,

$$\begin{aligned} \frac{W}{K} &> q \\ m^* \geq \frac{W}{K} &> q \\ m^* \geq \frac{W}{K} &\geq q + 1 \\ m^* &\geq q + 1 \\ 2m^* &\geq 2q + 2 > 2q + 1 = m \\ m &< 2m^* \end{aligned}$$

In both cases we have a guarantee on the maximum number of trucks being used as double that of the optimal solution.

Problem 2. *3-D Matching.*

Solution. Use this algorithm:
Iterate through T and add the current triple if no member of the current triple is already in M. Output M. This runs in $O(|T|)$ time.

This algorithm clearly gives a feasible solution because it relies on the definition of the problem to decide whether or not to include any given triple.

This algorithm is clearly poly-time in $|T|$.

Let M^* be the OPT solution. In the form, $(a_i, b_i, c_i) \forall i \in M^*$, at least one of a_i, b_i, c_i must appear in greedy solution M or else a_i, b_i, c_i would be in M. All members of the triple can appear in M at most once. So every triple in M can "pay for" at most 3 triples in M^*. Thus M^* cannot have more than 3 times as many triples as M.

Problem 3. *Vertex Cover.*

Solution. This algorithm must be (1) feasible, (2) poly-time, and (3) have a guarantee.

(1) The solution is feasible. If DFS traverse edge $e = (u, v)$ and backtracked after reaching v, then either v either has no edges other than e or all neighbours of v have been visited. The neighbours of v cannot be leaf nodes, otherwise DFS could have reached v already. Ignoring the leaf nodes would give a vertex cover of G.

(2) DFS is poly-time so this algorithm is poly-time.

(3) There exists a matching $M \in G$ such that $|M| \geq \frac{1}{2}|W|$ where W is the DFS tree without leaf nodes. We can prove this by running DFS and colouring the current edge $e = (u, v)$ if neither u nor v have coloured edges. Except for edges where one vertex is a leaf node, all edges can be paired with an edges that came after it in DFS order such that one edge in the pair is coloured and one is not. Thus the number of coloured edges form a matching M and $|M| \geq \frac{1}{2}|W|$. M is also the size of optimal solution W^* because some set that chooses one of every terminal node in M is also a vertex cover in G. So, $|W^*| \geq \frac{1}{2}|W|$ and thus, $2 \cdot |W^*| \geq |W|$.

Problem 4. *Travelling Salesman Problem: Greedy Fit.*

Solution. Begin with the assumption that G is complete.

(a) Say $|H^*| \leq |M|$. There is a trivial case when M has more than 2 leaf nodes so adding the cycle constraint must cost more than M. When M has 2 leaf nodes, it is a Hamiltonian path. Adding an edge to M to create the tour H^* must increase the cost because this problem assume edge weights are

non-negative. But we assumed $|H^*| \leq |M|$. Contradiction. Thus $|H^*| > |M|$.

(b) (v_j, v_i) has to be greater than the distance between any previous node and T_{i-1}. We know that $|(v_j, v_i, v_{j+1})| > (v_j, v_{j+1})$ and $(v_j, v_i) + (v_i, v_{j+1}) > (v_j, v_{j+1})$ because of the triangle inequality.

Problem 5. *Travelling Salesman Problem: Greedy Walk.*

Solution. (a) Let P^* be the OPT solution. For every edge $e_i = (u_i, v_i) \in M$ add edge between v_i and u_j in $e_j = (u_j, v_j) \in M$ where u_j is closest to v_i. This is the optimal solution. e_i was chosen for M over (v_i, u_j) because $(u_i, v_i) \leq (v_i, u_j)$. There are twice as many edges in P^* as there are in M, so $|P^*| \geq 2|M|$.

(b) Let H be the OPT solution. M is the minimum cost perfect matching in G. Assume there is no edge from v_{2i} or v_{2i+1} that is at most c_{e_i}, i.e. all edges e_j from v_{2i} and v_{2i+1} have $c_{e_j} > c_{e_i}$. Then e_i cannot exist because e_i has c_{e_i}. Contradiction. There must be such an edge.

Problem 6. *Facility Location.*

Solution.