COMS E6232 - Problem Set #2

Alex Wong (asw2181) - asw2181@columbia.edu

March 3, 2017

Problem 1

a. Consider an instance where we have two items x and y where $s_x = \epsilon$, $v_x = 2\epsilon$, $s_y = B$, $v_y = B$, and $\epsilon << 1$. The v_i/s_i ratio of item x is 2 and item y is 1. Thus, the Greedy algorithm will always pick item x regardless of how large B is and regardless how small ϵ is. So as B gets larger and/or ϵ gets smaller, the approximation ratio of the algorithm will always increase, thus showing that the approximation ratio of Greedy is not bounded by any constant. □

b. **Theorem 1b** The Modified Greedy algorithm achieves approximation ratio 2.

Proof: Let OPT be the optimal solution that has maximum value where $v(OPT) = \sum_{i \epsilon OPT} v_i$ subject to $\sum_{i \epsilon OPT} s_i \leq B$. We first assume that the items are ordered in a non-increasing fashion according to the ratio v_i/s_i. Let's call the first item that does not fit in the knapsack using the Greedy algorithm as item m. We know that item m's v_i/s_i ratio $\geq$ items $m+1, ..., n$'s v_i/s_i ratio. Thus, if we are able to fit some fraction of item m so that it fills up to the capacity of the knapsack, that solution will be $\geq$ OPT. We can define the fraction of item m that fits into the knapsack as α where $\alpha = (B - \sum_{i=1}^{m-1} v_i)/s_m$. Thus, $OPT \leq (\sum_{i=1}^{m-1} v_i) + \alpha v_m$. Since α is some fraction ≤ 1, we can also say that:

$$OPT \leq (\sum_{i=1}^{m-1} v_i) + \alpha v_m \leq (\sum_{i=1}^{m-1} v_i) + v_m$$

From the inequality above, $\sum_{i=1}^{m-1} v_i$ or v_m must be at least $OPT/2$, showing that the Modified Greedy algorithm will always get a solution at least $OPT/2$, thus achieving an approximation ratio 2. □

Problem 2

a. We know that the lower bound for OPT is $max(max_i(p_i), \frac{\sum_i p_i}{m})$ where $i = 1, ..., n$.

For $m = 2$ machines and 5 jobs with processing times 3,3,2,2,2, the LPT algorithm will schedule 3,2,2 on m_1 and 3,2 on m_2, giving a makespan of 7. For OPT, we know that a lower bound is max $= max(3, 12/2) = 6$. We can achieve OPT by scheduling 3,3 on one machine and 2,2,2 on the other machine.

For $m = 3$ machines and 7 jobs with processing times 5,5,4,4,3,3,3, the LPT algorithm will schedule 5,3,3 on m_1, 5,3 on m_2, and 4,4 on m_3, giving a makespan of 11. For OPT, the lower bound is $max(5, 27/3) = 9$. We can achieve OPT by scheduling 5,4 on one machine, 5,4 on another machine, and 3,3,3 on the remaining machine.

2. **Lemma 2.2** Let U be the subset of nodes consisting of the nodes of $\{1, ..., n\}$-$\{s, t\}$ that have odd degree in T and the nodes in $\{s, t\}$ that have even degree in T. Then U has an even number of nodes.

Proof: Suppose we partition U into 2 subsets: U_{odd} and U_{even} where they contain nodes that have odd degree and nodes that have even degree, respectively. We know that in the minimum spanning tree T, there is an even number of nodes that have odd degree since $\sum_{v \in T_{odd}} degree(v) = 2 \cdot \#edges$. Knowing this, if $|U_{odd}|$ is odd, that means either s or t must have an odd degree, meaning the other must have an even degree and would be contained in U_{even}, so $|U_{even}| = 1$. Thus,

$$|U| = |U_{odd}| + |U_{even}| = odd\# + 1 = even\#$$

which shows that $|U|$ is even.

If $|U_{odd}|$ is even, then s and t must both have odd degrees or both have even degrees; if both are odd, then $|U_{even}| = 0$ which trivially shows $|U|$ is even. If both are even, then $|U_{even}| = 2$ and

$$|U| = |U_{odd}| + |U_{even}| = even\# + 2 = even\#$$

which shows that $|U|$ is even. It must be noted that if $|U_{even}| = 2$ there must be other nodes in set U; U can never just contain s and t. Thus, the set U has an even number of nodes. $\square$

3. **Lemma 2.3** Let $G[U]$ be the subgraph of G induced by the subset U, M be a minimum-cost perfect matching in $G[U]$, and P^* be a minimum-cost path from s to t that visits every node exactly once. Consider the nodes of U in the order that they appear on the path P^* and let u_i be the i-th node of U in this ordering: $i = 1, ..., |U|$. Color red the edges of P^* on the subpath from u_1 to u_2,..., from u_{2k-1} to u_{2k}, where

$|U| = 2k$. Color blue the other edges of P^*. Then, cost(red edges) $\geq$ cost(M).

Proof: We can prove this by contradiction. Suppose cost(red edges) $<$ cost(M). We know that red edges are essentially a perfect matching on set U. If cost(red edges) $<$ cost(M), then M is not a minimum-cost perfect matching on set U, proving that cost(red edges) $<$ cost(M) can not be true. Thus, cost(red edges) $\geq$ cost(M). □

4. **Lemma 2.4** cost(T) + cost(blue edges) $\geq 2 \cdot$ cost(M).

Proof: We first see that $T \cup$ (blue edges of P^*) creates a connected Eulerian multigraph. We know this to be true because every vertex has an even degree. A Eulerian multigraph is essentially a multigraph that contains an Eulerian tour. Within a Eulerian tour, we know that we can create two disjoint perfect matchings of U and the sum of their costs $\leq$ cost of the Eulerian tour due to triangle inequality. We also know that both of the disjoint perfect matchings can not have a cost $<$ cost(M) or else M can't be a minimum-cost perfect matching on U. Thus,

$$2 \cdot cost(M) \leq cost(2\ disjoint\ perfect\ matchings) \leq cost(T) + cost(blue\ edges)$$

$$cost(T) + cost(blue\ edges) \geq 2 \cdot cost(M)$$

□

5. **Theorem 2.5** The variant of Christofides' algorithm achieves an approximation factor of 5/3 for the *s-t path metric TSP* problem.

Proof: Combining our conclusions from Lemma 2.3 and 2.4, we can say that:

$$cost(T) + cost(blue\ edges) + cost(red\ edges) \geq 3 \cdot cost(M) \tag{1}$$

We also know that $cost(blue\ edges) + cost(red\ edges) = P^* = OPT$ and from Lemma 2.1 that $cost(T) \leq OPT$. Thus, we can modify equation 1 to be:

$$cost(T) + OPT \geq 3 \cdot cost(M)$$

$$2 \cdot OPT \geq 3 \cdot cost(M)$$

$$cost(M) \leq 2/3\ OPT \tag{2}$$

Adding cost(T) to both sides of equation 2 and using Lemma 2.1 that $cost(T) \leq OPT$, we finally get:

$$cost(T) + cost(M) \leq cost(T) + 2/3\ OPT$$

$$cost(T) + cost(M) \leq OPT + 2/3\ OPT$$

$$cost(T) + cost(M) \leq 5/3\ OPT$$

Thus showing that the cost of the path computed by the variant of Christofides' algorithm is at most 5/3 times the cost of the optimal path. □

6.