601.435 / 601.635 Approximation Algorithms **Spring 2026**
Homework #1 **Due:** February 12, 2026, 9:00am

Reminder: you may work in groups of up to three people, but must write up solutions entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. Many of these problems have solutions which can be found on the internet – please don't look. You can of course use the internet (including the links provided on the course webpage) as a learning tool, but don't go looking for solutions.

Please include proofs with all of your answers, unless stated otherwise.

1 k-suppliers (33 points)

The k-suppliers problem is similar to k-center. We are given a metric space (V, d) and a natural number k. However, in k-suppliers the set of points V is partitioned into two sets: the *suppliers* F and the *customers* $D = V \setminus F$. The goal is to find a set of suppliers $S \subseteq F$ with $|S| = k$ that minimizes $\max_{u \in D} d(u, S)$. Give a 3-approximation algorithm for this problem. Hint: think about the greedy 2-approximation for k-center from class

Solution: There are a few ways of doing this. Here's one. For each $v \in D$, let $f(v) \in F$ be the supplier closest to v. We use our greedy 2-approximation just on D to get a set D' (with $|D'| = k$), and then we let $S = \{f(v) : v \in D'\}$. If $|S| < k$, then we add $k - |S|$ suppliers to S arbitrarily.

Clearly S is feasible and runs in polynomial time, so it remains to prove the approximation ratio. Let $S^* \subseteq F$ be the optimal solution. We claim that $d(v, D') \leq 2 \cdot OPT$. To see this, suppose that it is false: $d(v, D') > 2 \cdot OPT$. Then since the greedy algorithm always adds the node which is furthest away from the current set, it must be the case that $d(a, b) > 2 \cdot OPT$ for all $a, b \in D'$. Then $D' \cup \{v\}$ is a set of size $k + 1$ where every two elements are at least $2 \cdot OPT$ away from each other. This means that for every $u \in S^*$, there is at most one vertex from $D' \cup \{v\}$ at distance at most OPT from u: if there were two such nodes, then by triangle inequality they would at distance at most $2 \cdot OPT$ from each other. But this implies that $|S^*| \geq k + 1$, which is a contradiction.

Thus $d(v, D') \leq 2 \cdot OPT$. Let $v' \in D'$ be the node in D' closest to v. So $d(v, S) \leq d(v, v') + d(v', f(v')) \leq 2 \cdot OPT + OPT = 3 \cdot OPT$.

2 Hardness of Minimum Degree Spanning Tree (33 points)

In the minimum degree spanning tree (MDST) problem we are given a graph $G = (V, E)$, and are asked to find the spanning tree which minimizes the maximum degree. Prove that unless P = NP, there is no α-approximation for this problem with $\alpha < 3/2$.

Hint: consider the Hamiltonian Path problem, which is known to be NP-hard. If there is a Hamiltonian path, what does that imply about the MDST? If there is no Hamiltonian path, what does that imply aout the MDST?

Solution: We reduce from Hamiltonian Path. Suppose that there were an algorithm $\mathcal{A}$ for MDST with approximation ratio $\alpha < 3/2$. Then given an instance $G = (V, E)$ of Hamiltonian Path, we run $\mathcal{A}$ on G and return YES if it returns a value of at most 2 (i.e., a tree with maximum degree at most 2) and return NO if it returns a value of at least 3.

Suppose that G has a Hamiltonian path. Then since $\mathcal{A}$ is an $\alpha < 3/2$ approximation, it will return a tree with maximum degree at most $\alpha \cdot 2 < 3$, so since degrees are always integral it must return a tree with maximum degree at most 2. So we will return YES. On the other hand, suppose that G does not have a Hamiltonian path. Then there is no spanning tree with maximum degree less than 3, so $\mathcal{A}$ will return a solution with maximum degree at least 3, and we will return NO. Thus we would have a polynomial-time algorithm for Hamiltonian Path.

3 Edge-Disjoint Paths (34 points)

In the edge-disjoint paths problem (EDP), the input is an undirected graph $G = (V, E)$ and a set $T = \{(s_1, t_2), (s_2, t_2), \ldots, (s_k, t_k)\}$, such that $s_i, t_i \in V$ for all $i \in [k]$. A feasible solution is a set $I \subseteq [k]$ and for all $i \in I$ a path P_i between s_i and t_i, with the additional constraint that $P_i \cap P_j = \emptyset$ for $i, j \in I$ with $i \neq j$ (where we view paths as edge sets). In other words, a feasible solution is a set of edge-disjoint paths between a subset of the pairs in T. The objective is to maximize $|I|$, i.e. the number of edge-disjoint paths that we can find.

Consider the following greedy algorithm, where initially $I \leftarrow \emptyset$:

1. Repeat until all pairs (s_i, t_i), $i \notin I$ are disconnected in G:
   (a) Let $i^* \in [k] \setminus I$ be the index which minimizes the distance between s_i and t_i in G, i.e. i^* is the index of the closest pair in G.
   (b) Let P_{i^*} be a shortest path between s_i and t_i in G.
   (c) Add i^* to I and choose path P_{i^*} for i^*, and remove all edges of P_{i^*} from G.
2. Return I

Informally, this algorithm just always picks the shortest possible path remaining, then deletes this path from the graph and continues. Prove that this is an $O(\sqrt{m})$-approximation (where $m = |E|$).

Hint: divide paths up into short paths (length at most $\sqrt{m}$) and long paths (length larger than $\sqrt{m}$).

Solution: Let $I^* \subseteq [k]$ be the optimal solution with associated paths P_i^* for all $i \in I^*$. Let $\mathcal{P}^* = \{P_i^* : i \in I^*\}$ be the optimal set of paths. Let I and $\mathcal{P}$ be the solution generated by the greedy algorithms. For every $i \in I^* \setminus I$, the only reason that $i \notin I$ is because at least one edge of P_i^* intersects some P_j with $j \in I \setminus I^*$ and $|P_j| \leq |P_i^*|$. In other words, the only reason the greedy algorithm couldn't choose path P_i^* is because at least one edge was already used by a path that the greedy algorithm picked earlier. For every $i \in I^* \setminus I$, let $\Phi(i)$ denote such a $j \in I$. We think of j as "charging" i.

Let $I_{short} = \{i \in I \setminus I^* : |P_i| \leq \sqrt{m}\|$, and let $I_{long} = \{i \in I \setminus I^* : |P_i| > \sqrt{m}\}$. Since every

$i \in I^* \setminus I$ charges some element of $I \setminus I^*$, we know that

$$\begin{aligned}
|I^* \setminus I| &= \sum_{j \in I \setminus I^*} |\{i \in I^* \setminus I : \Phi(i) = j\}| \\
&= \sum_{j \in I_{short}} |\{i \in I^* \setminus I : \Phi(i) = j\}| + \sum_{j \in I_{long}} |\{i \in I^* \setminus I : \Phi(i) = j\}|
\end{aligned}$$

We can now bound these terms separately. For the first term, note that all paths in $\mathcal{P}^*$ are edge-disjoint, so there can be at most $|P_j|$ paths from $\mathcal{P}^*$ which share at least one edge with P_j. Thus $|\{i \in I^* \setminus I : \Phi(i) = j\}| \leq |P_j|$ for all $j \in I \setminus I^*$. Thus

$$\sum_{j \in I_{short}} |\{i \in I^* \setminus I : \Phi(i) = j\}| \leq \sum_{j \in I_{short}} |P_j| \leq |I_{short}|\sqrt{m}$$

On the other hand, for the second term, recall that every $i \in I^* \setminus I$ with $\Phi(i) = j$ must have $|P_i^*| \geq |P_j|$ (paths in the optimal solution charge shorter paths that were already chosen by greedy). So if $j \in I_{long}$, every $i \in I^* \setminus I$ with $\Phi(i) = j$ must have $|P_i^*| \geq |P_j| \geq \sqrt{m}$. Clearly there can only be at most $\sqrt{m}$ edge disjoint paths of length at least $\sqrt{m}$ (since there are only m edges total), so

$$\sum_{j \in I_{long}} |\{i \in I^* \setminus I : \Phi(i) = j\}| \leq \sqrt{m}$$

Putting this all together, we get that

$$\begin{aligned}
|I^*| &= |I^* \cap I| + |I^* \setminus I| \\
&\leq |I^* \cap I| + \sum_{j \in I_{short}} |\{i \in I^* \setminus I : \Phi(i) = j\}| + \sum_{j \in I_{long}} |\{i \in I^* \setminus I : \Phi(i) = j\}| \\
&\leq |I^* \cap I| + |I_{short}|\sqrt{m} + \sqrt{m} \\
&\leq |I| + |I|\sqrt{m} + |I|\sqrt{m} \\
&= O(|I|\sqrt{m})
\end{aligned}$$

Thus by definition the algorithm is an $O(\sqrt{m})$-approximation.