Fall 2021, CS 583: Approximation Algorithms

Homework 5

Due: 11/30/2021 in Gradescope

Instructions and Policy: Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people. You may be able to find solutions to the problems in various papers and books but it would defeat the purpose of learning if copy them. You should cite all sources that you use and write in your own words.

Read through all the problems and think about them and how they relate to what we covered in the lectures. Solve as many problems as you can. Please submit solutions to Problem 1, 6 and at least two other problems. Some problems are closely related so it may benefit you to solve them together or view them as parts of an extended problem.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.

Problem 1 We have mostly seen edge-weighted network design problems in undirected graphs. As we discussed in lecture, directed network design problems are hard to approximate. Undirected *node-weighted* network design problems fall in between. Here we consider node-weighted Steiner tree problem. The input is a graph $G = (V, E)$ with non-negative node weights $w : V \to \mathbb{R}_+$, and a set of k terminals $S \subseteq V$. The goal is to find a Steiner tree $T = (X, F)$ for the terminals (that is, $S \subseteq X$) with minimum node weight. We will assume that the weight of the terminals is 0 without loss of generality since every terminal has to be included in the tree. We saw that Steiner tree in edge-weighted graphs admits a simple 2-approximation via the MST heuristic, and improved approximations are also known. In contrast the node-weighted problem is harder. This problem will walk you through an approach to approximate it via Set Cover type ideas.

- Describe a reduction from Set Cover on m sets and n elements to node-weighted Steiner tree with $k = n$. Informally argue that an $\alpha(k)$-approximation for node-weighted Steiner tree implies an $\alpha(n)$-approximation for Set Cover. This should convince you that node-weighted Steiner tree does not admit a better than $\ln k$ approximation.
- Given a set of terminals, a spider is a star-like tree with a center vertex c and k legs where each leg is a c-t path for some terminal t (the center itself can be a terminal). Thus, all the leaves are terminals if $k > 1$. If $k = 1$ then we require that the center to be a terminal and in this case the spider is simply a path connecting two terminals. See figure. Argue that any minimal Steiner tree (X, F) contains a collection of node-disjoint spiders that together include all but one terminal from S.

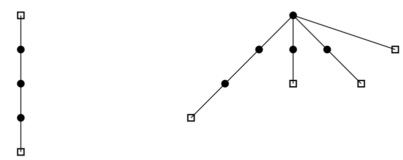

Figure 1: Examples of spiders with $k = 1$ and $k = 4$. Terminals are shown as a box and non-terminals as a circle.

- The density of a spider is the ratio of the total weight of the nodes in the spider divided by the number of terminals in it. Describe a polynomial-time algorithm to find the best density spider.

- Consider the following algorithm. It finds a best density spider and contracts the nodes of the spider to the center which now becomes a terminal, and recurses on the residual instance. Formalize this algorithm and use Set Cover like analysis to argue that it yields a $O(\ln k)$ approximation.

Problem 2 We discussed the Point-to-Point connection problem and mentioned that it can be cast as a special case of covering a proper function. See survey of Goemans and Williamson on network design. Here we consider the *unbalanced* point-to-point connection problem. The input consists of an edge-weighted undirected graph $G = (V, E)$ and two disjoint sets of vertices S and T where $|S| \leq |T|$. The goal is to find a min-cost subgraph H of G such that in each connected component C of H there are at least as many nodes from T as there are from S; that is $|V(C) \cap S| \leq |V(C) \cap T|$. When $|S| = |T|$ we have the Point-to-Point connection problem. Show that if $|S| < |T|$ the resulting problem is not necessarily a special case of covering a proper or the more general problem of covering an uncrossable function.

Problem 3 Problem 23.23 in Vazirani's book.

Problem 4 Problem 11.3 from Williamson-Shmoys book.

Problem 5 Problem 11.5 from Williamson-Shmoys book.

Problem 6 In the Node-weighted Multiway Cut problem we are given an undirected node-weighted graph $G = (V, E)$, k terminal nodes $S = \{s_1, s_2, \ldots, s_k\}$. Node v has a non-negative weight c_v. The goal is to remove a minimum weight subset of nodes $V' \subset V$ such that $G - V'$ has no path from s_i to s_j, $1 \leq i < j \leq k$. Assume for simplicity that terminals cannot be removed (they have infinite weight) and that they form an independent set (so that there is a feasible solution). Consider the following LP relaxation where there is a variable x_v for each $v \in V \setminus S$ indicating whether to remove v. Let $\mathcal{P}_{u,v}$ denote the set of all paths from u to v in G.

$$
\begin{aligned}
\min \sum_{v \in V} c_v x_v & & & \\
\sum_{v \in p} x_v &\geq 1 & & p \in \mathcal{P}_{s_i,s_j}, i \neq j \\
x_v &\geq 0 & & v \in V
\end{aligned}
$$

Let $\bar{x}$ be a feasible solution to the LP. Define $B_{\bar{x}}(s,r)$ where $s \in V$ and r is a real number to be the set of all nodes v such that there is a path P from s to v such that $(\sum_{u \in P} \bar{x}_u) < r$.

Consider the following rounding algorithm. First remove all nodes v such that $\bar{x}_v \geq 1/3$. Second, pick a θ uniformly at random from $(0, 1/3)$ and for each s_i remove all nodes that are adjacent to $B(s_i, \theta)$ but not in $B(s_i, \theta)$. Prove that for any θ the removed nodes form a multiway cut and that the expected weight of the nodes removed is at most $3 \sum_v c_v x_v$.

Note that this problem admits a 2-approximation via a reduction to the Directed Multiway Cut problem that we saw in lecture. The problem is designed to make you work with θ-rounding for cut problems and node-weighted problems.

Problem 7 A *hypergraph* $G = (V, \mathcal{E})$ consists of nodes V and hyperedges $\mathcal{E}$. Each hyperedge $e \in \mathcal{E}$ is a subset of V. The *rank* of a hypergraph G is defined as the maximum size of any of its edges, that is, $r = \max_{e \in \mathcal{E}} |e|$. Graphs are hypergraph with rank 2. In Hypergraph-Multiway-Cut the input is a hypergraph G, weights on the hyperedges $w : \mathcal{E} \to \mathbb{R}_+$ and k terminals $\{s_1, \ldots, s_k\}$. The goal is to remove the minimum weight set of hyperedges from G such that for any $i \neq j$ s_i and s_j are *disconnected* (I assume you will be able to generalize the notion of connected from graphs to hypergraphs in the natural fashion).

- Consider the natural generalization of the Isolating-Cut heuristic that we saw for Graph Multiway-Cut. Show that it is a $r(1 - 1/k)$-approximation algorithm where r is the rank of the hypergraph. Also demonstrate that your analysis is tight in the worst case.
- Show that Hypergraph Multiway-Cut can be reduced to the Node-weighted Multiway Cut problem in an approximation preserving fashion. And deduce a 2-approximation.
- You can also reduce Node-weighted Multiway Cut to Hypergraph Multiway Cut. Do you see how? No need to write up this part.