Spring 2018, CS 583: Approximation Algorithms

Homework 6

Due: 05/04/2018

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.

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 your solutions to at least 4 problems.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary. Your job is to convince me that you know the solution, as quickly as possible.

Problem 1 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 2 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.

Problem 3 Consider the feedback edge set problem (FES). The input is an edge-weighted undirected graph $G = (V, E)$ and the goal is to remove a minimum-weight set $E' \subset E$ such that $G - E'$ has no cycles. Note that FES can be solved in polynomial-time by taking the complement of a maximum-weight spanning tree. Nevertheless we will consider an analysis based on the following natural LP. There is a variable x_e for each $e \in E$ that indicates whether to remove e.

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & & \\ \sum_{e \in C} x_e &\geq 1 & & \text{for each cycle } C \\ x_e &\geq 0 & & e \in E \end{aligned}$$

- Assuming the existence of a constant degree graphs whose *girth* is $\Omega(\log n)$ (the degree is bigger than 2 otherwise the cycle is an easy example) prove that the integrality gap of this LP is $\Omega(\log n)$. The girth of a graph is the length of its shortest cycle.
- We saw an upper bound of $O(\log n)$ via the primal-dual technique. See Section 7.2 in Williamson-Shmoys book for the more general Feedback Vertex Set problem. Here we will see a proof via a reduction to the Multicut analysis. Given a feasible solution $\bar{x}$ for the LP for FES, we first remove all edges e with $\bar{x}_e > 1/3$ and then define a Multicut

instance where each a pair of vertices uv is to be separated of $d_{\bar{x}}(uv) \geq 1/3$; that is, the distance from u to v according to edge-lenghts given by $\bar{x}$ is at least $1/3$. Define a feasible fractional solution for this Multicut instance on G by appropriately scaling up $\bar{x}$ and use the Multicut analysis to give an $O(\log n)$ upper bound on the integrality gap of the LP.

- *Extra Credit:* Use the above ideas to obtain an $O(\log n)$-approximation for the Subset Feedback Edgeset problem. Here we are given edge-weighted graph $G = (V, E)$ and a subset of terminals $S = \{s_1, \ldots, s_k\}$ and the goal is to remove a minimum-weight set of edges E' such that $G - E'$ has no cycle containing a terminal.

Problem 4 In the k-MST problem you are given an undirected edge-weighted graph $G = (V, E)$ with edge weights $c : E \to \mathbb{R}^+$ and an integer k. The goal is to find a tree $T = (V_T, E_T)$ in G of smallest edge weight ($\sum_{e \in E_T} c(e)$) such that $|V_T| \geq k$. Show that if there is an α-approximation for k-MST then there is an α-approximation for the Steiner tree problem. Recall that in the Steiner tree problem, the input is an edge-weighted graph $G = (V, E)$ and a set of terminals $S \subseteq V$; the goal is to find a tree T of minimum edge-weight that connects (contains) all the terminals S.

Problem 5 Here we consider the rooted k-Steiner problem which is related to the previous problem. The input consists of an edge-weighted undirected graph $G = (V, E)$, a specified root vertex r and a set $S \subset V$ of terminals. The goal is to find a min-cost tree (V_T, E_T), a sub-graph of G, such that $r \in V_T$ and $|S \cap V_T| \geq k$. Obtain a randomized $O(\log n)$ approximation via probabilistic embedding into tree metrics.

There is a 2-approximation for this problem but it is rather involved.

Problem 6 Prove that any ring metric isometrically embeds into ℓ_1.

Problem 7 Given a graph $G = (V, E)$ with edge-weights $c : E \to \mathbb{R}^+$, you wish to partition G into $G_1 = G[V_1], G_2 = G[V_2], G_3 = G[V_3]$ such that $\lfloor |V|/3 \rfloor \leq |V_i| \leq \lceil |V|/3 \rceil$ for $1 \leq i \leq 3$, and the cost of the edges between the partitions is minimized. Using an α-approximation for the sparsest cut problem, give a pseudo-approximation for this problem where you partition the graph into 3 pieces $G[V_1'], G[V_2'], G[V_3']$ such that $|V|/c_2 \leq |V_i'| \leq |V|/c_1$ for some constants $1 < c_1 < c_2$ and the cost of the edges between the partitions is $O(\alpha)\text{OPT}$. What constants c_1, c_2 can you guarantee? Note that c_1 and c_2 should be *constants*, independent of the graph size. (Hint: this problem is similar to the one on partitioning into two pieces that is in Vazirani's book on applications of sparsest cut (Section 21.6.3).)

Problem 8 Consider MAX-CUT with the additional constraint that specified pairs of vertices be on the same/opposite sides of the cut. Formally, we are given two sets of pairs of vertices, S_1 and S_2. The pairs in S_1 need to be separated, and those in S_2 need to be on the same side of the cut sought. Under these constraints, the problem is to find a maximum-weight cut.

1. Give an efficient algorithm to check if there is a *feasible* solution.

2. Assuming there is a feasible solution, give a strict quadratic program and vector program relaxation for this problem. Show how the algorithm for MAX-CUT we saw in class can be adapted to this problem while maintaining the same approximation ratio.

Problem 9 Problem 6.6 from the Williamson-Shmoys book. SDP for directed cut.