Fall 2024, CS 598: Topics in Graph Algorithms
Homework 4
Due: 12/12/2024

Instructions and Policy: You can work in groups of up to two. Each group needs to submit only one solution. You need to indicate the names of the people you discussed a problem with. Solutions to most of these problems can be found from one source or the other. Try to solve on your own first, and cite your sources if you do use them.

Please write clearly and concisely. Refer to known facts. You should try to convince me that you know the solution, as quickly as possible.

For this home work submit solutions to any three problems.

Problem 1 Recent work has shown that min-cost s-t flow can be solved very fast, in almost linear time when capacities are polynomially-bounded. Can we use min-cost flow as an oracle to approximately solve multicommodity flow? The goal of this problem is to use MWU for this purpose. Suppose we want to solve max multicommodity flow in a directed graph $G = (V, E)$ with k pairs $(s_1, t_1), \ldots, (s_k, t_k)$ and non-negative integer capacities $u(e), e \in E$. For each pair i, consider the polytope $\mathcal{P}_i$ which is the set of all s_i-t_i flows (with edge variable $f_i(e), e \in E$ that satisfy the capacity constraints. Let $\mathcal{P} = \prod_i \mathcal{P}_i$ be the polytope obtained by considering the k flows separately. Note that the dimension of $\mathcal{P}$ is mk and any point $x \in \mathcal{P}$ corresponds to k flows, one for each pair i, such that individually the flows are feasible in G for that pair.

- Write the max multicommodity flow problem as an implicit LP where the goal is to find a point $x \in \mathcal{P}$ that satisfies the total capacity constraint for each edge and with an appropriate objective. Ensure that the objective is set up so that this is a packing LP.
- What is the *width* of the LP in terms of k?
- Use MWU for packing with width-dependent algorithm to obtain a $(1 - \epsilon)$-approximation for the max-multicommodity flow problem that runs in $O(\frac{k}{\epsilon^2} T(n, m) \log m)$ where $T(n, m)$ is the running time for s-t mincost flow in a n-node m-edge graph.

Problem 2 Let $G = (V, E)$ be a graph with non-negative edge-costs $c(e), e \in E$. We will consider the 2ECSS problem and an LP relaxation for it. In this problem the goal is to choose a min-cost subset of edges $E' \subseteq E$ such that the graph $H = (V, E')$ is two edge connected. In fact we will allow an edge e to be chosen twice (the version where we can only choose one copy is actually more interesting but we will not worry about that for now).

- Using a variable x_e for each edge e write a simple covering LP relaxation for the problem which has an exponential number of constraints, one for each cut in the graph. Write down its dual as a packing LP with an exponential number of variables but a polynomial number of constraints.

- Show how to use the MWU method for implicit packing LPs to derive a polynomial time approximation scheme for solving the dual LP. What is the oracle and what is the total running time to obtain a $(1-\epsilon)$-approximation for the LP.

- Recall that we actually want to compute an approximate solution to the primal LP while the MWU method is approximately solving the dual LP. Argue that one can obtain an approximate solution to the primal LP as follows. Consider the weights maintained by the MWU algorithm while solving the dual LP. Show that there is some iteration of the algorithm such that the weights in that iteration form a $(1+O(\epsilon))$-approximation to the primal LP.

Problem 3 For a graph $G=(V,E)$ a set of vertices $S \subseteq V$ is a vertex cover of G if every edge of E intersects S. The min-vertex-cover problem is to find a vertex cover in G of smallest cardinality. This is an NP-Hard problem.

- Suppose G has treewidth k and you are given a tree decomposition of width k. Describe a $2^{O(k)}\text{poly}(n)$-time algorithm to solve the min-vertex-cover problem in G.

- **Extra credit:** We stated a theorem in lecture that if $G=(V,E)$ is a planar graph then for any $h>1$ there is an efficient algorithm to color V with h colors such that for any $i \in [h]$, $G[V-V_i]$ has treewidth at most ch for some fixed constant c. A similar theorem also holds where the edges are partitioned into h color classes $E_1,\ldots,E_h$ and the induced graph $G_i = G[E-E_i]$ has treewidth at most ch. Using this algorithm, and assuming that you can find a tree decomposition of a graph efficiently, describe an algorithm that given a planar graph G and any fixed $\epsilon>0$, outputs a minimum vertex cover in time $2^{O(1/\epsilon)}\text{poly}(n)$ time.

Problem 4 Let $G=(V,E)$ be a directed graph. Given a subset $X \subseteq V$ we say that X is well-linked if for any $A,B \subseteq X$ with $|A|=|B|$ and $A \cap B = \emptyset$ there are $|A|$ edge-disjoint paths from A to B just as we defined in undirected graphs. Note that we can swap A and B so we also have paths from B to A. Describe a cut-matching game following KRV style one that either proves that X is not well-linked or outputs a certificate that X is well-linked in G with congestion $O(\log^2|X|)$. This gives a fast approximation algorithm for the directed sparsest cut problem.