Fall 2024, CS 598: Topics in Graph Algorithms
Homework 3
Due: 11/06/2024

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 five of the eight problems.

Problem 1 We saw expander decomposition in the lecture where we discussed it for decomposing such that each cluster has good conductance. In the notes, Theorem 14 sketched how the same procedure can be applied to more general settings. Using these ideas solve Problem 1 in Sepher Assadi's HW 2.

Problem 2 Let $G = (V, E)$ be an undirected graph with non-negative edge costs $c : E \to \mathbb{Z}_+$. Suppose $n = |V|$ is even. A bisection is a partition of V into V_1 and V_2 with $|V_1| = |V_2| = n/2$. The goal in min-bisection is to find a bisection that minimizes the number of edges crossing the partition. A relaxation of a min-bisection is a $3/4$-balanced partition where we have the property that $|V_1|$ and $|V_2$ are at most $3n/4$. The goal is again to find a $3/4$-balanced partition that minimizes the number of edges crossing the cut. Such balanced partitions are very useful in divide-and-conquer algorithms and are the work-horse of graph partitioning for parallel numerical computation. In fact those applications were some of the initial reasons to care about sparsest cut.

- Suppose we have an α-approximation for Uniform Sparsest Cut. Describe an algorithm that outputs a $3/4$-balanced partition whose cost is at most $O(\alpha)$OPT where OPT is the cost of an optimum bisection. This is called a pseudo-approximation since we are comparing with the cost of a solution which can potentially be much larger than the cost of a $3/4$-balanced partition; can you think of an example?
- Obtaining an approximation algorithm for the exact bisection is harder. To see why, describe an algorithm that decides whethere OPT $= 0$.
- Describe how you would solve the problem when G is a tree.
- Use tree-based probabilistic oblivious routing (assume that you can sample from the distribution in polynomial time) to obtain an $O(\log n)$-approximation for the min-bisection problem.

Problem 3 Solve Exercise 4 in the lecture notes on Räcke's cut-based hierarchical decomposition.

Problem 4 Kent discussed proper h-hop distances which are hard to compute but can be done efficiently for small h via the powerful randomization technique called color coding. Let $G = (V, E)$ be a directed graph with edge weights $w : E \to \mathbb{R}$. Given a source s and a sink t and an integer h we wish to find a shortest s-t walk that contains exactly h negative weight edges which are distinct on the walk. We call this proper h-hop s-t distance.

- Prove that finding proper h-hop s-t distance is NP-Hard.
- Suppose the negative weight edges are assigned colors from $\{1, 2, \ldots, h\}$. Describe an algorithm to find the proper h-hop distance that contains exactly h negative weight edges of different colors (and hence necessarily distinct) in $O(2^h \text{poly}(n, m))$ time.
- Suppose we want to solve the original problem of proper h-hop distance. Describe a randomized algorithm that runs in $O(2^{O(h)} \text{poly}(n, m))$ time.

Problem 5 We discussed low-diameter decomposition for directed graphs in the context of the new shortest path algorithm. As mentioned in the lecture, directed graph decomposition was initially useful for cut and flow problems. We will illustrate this in the following problem. Let $G = (V, E)$ be a directed graph with non-negative edge costs $c : E \to \mathbb{Z}_+$. In the Feedback Arcset Problem (FAS)[1] the goal is to remove a min-cost subset of edges E' such that $G - E'$ is acyclic (a DAG).

- Write an LP relaxation with edge variables $x_e, e \in E$ and an exponential number of constraints, one for each cycle in G. Briefly describe why there is an efficient separation algorithm for the problem.
- Use the low-diameter decomposition result discussed in lecture to derive an $O(\log^2 n)$-approximation for FAS by rounding a fractional solution.

Problem 6 This is essentially a generalization of the previous problem. We have seen multicommodity flows and cuts in undirected graphs and saw that the flow-cut gaps are bounded by $O(\log n)$ for both Multicut and Sparsest Cut. However the flow-cut gap in directed graphs are $\Omega(n^\delta)$ in the worst-case. Nevertheless, for *symmetric* demands the flow-cut gap is poly-logarithmic and this can be seen via low-diameter decompositions. The goal of the problem is to make this connection. Let $G = (V, E)$ be directed graph with non-negative edge costs $c : E \to \mathbb{R}_+$. Consider a set of k unordered source-sink pairs $s_1 t_1, \ldots, s_k t_k$. We think of cuts as edge sets. We say a cut-set $E' \subseteq E$ disconnects a pair $s_i t_i$ if s_i and t_i are in different strong connected components of $G - E'$; one way to think about this is that we want to make sure that at least one of s_i, t_i cannot reach the other and we do not care which one. A Multicut for a given instance of symmetric demands in directed graphs is a set of edges E' such that all the input pairs are separated by the cut E'.

[1] In some settings people refer to edges in directed graphs as arcs. That is the reason for this name.

- Use low-diameter decomposition to obtain an $O(\log^2 n)$-approximation for Multicut for symmetric demands. Why does this generalize the algorithm for FAS in the previous problem?
- The sparsity of a cut E' is the ratio of $c(E')$ to the number of pairs separated by E'. Unlike the case of undirected graphs where we used embeddings, there is no clean way to obtain a good algorithm for Sparsest cut but one can use the scaling trick via Multicut. Read Section 3 in the lecture notes on Sparsest Cut and adapt the ideas to obtain an $O(\log^3 n)$ approximation for Sparsest Cut in the directed setting for symmetric demands.

Problem 7 Recall that the LDD we saw in lecture gave us a bound of $O(\log^2 n)\frac{w(e)}{D}$ on the probability of cutting an edge of weight/length $w(e)$ while guaranteeing that the diameter of each remaining strongly connected component is at most D. Can we improve this bound? A lower bound is $\Omega(\log n)$ which comes from undirected graphs. We however know that the LP integrality gap for a natural LP relaxation for the Multicut problem with symmetric demands that we discussed in the previous problem is $O(\log n \log\log n)$. Using this fact and duality argue that there is an LDD such that the cutting probability is $O(\log n \log\log n)\frac{w(e)}{D}$. You may want to see the lecture notes on LDDs for undirected graphs which gives the duality argument.

Problem 8 A key component of the fast scaling based SSSP algorithms is the fast randomized low-diameter decomposition. We did not have time to discuss the actual algorithm in the lecture. Read one of two recent shortest path papers that describe algorithms for ldd and describe it in your own words.