Advanced Algorithms 10/23, 2020

Exercise 05

Lecturer: Mohsen Ghaffari *Teaching Assistant: Vaclav Rozhon*

1 Tree embedding of cycle (leftover)

In this exercise, we consider the tree embedding for the simple case where the graph is just a cycle on n vertices. That is, the vertices of our graph are numbers $1, 2, \ldots, n$ and there is an edge between each i ($1 \leq i \leq n-1$) and $i+1$, as well as between n and 1. Recall that we wish to approximate the metric induced on our graph by shortest paths (that is, distance between $i < j$ is $d_G(i,j) = \min(|j-i|, |n-j+i|)$) by a metric d_T induced by shortest paths on a weighted tree T with the same vertex set as in our cycle.

In the next two parts, first, we argue that we cannot hope for a good deterministic solution. Then, we show that for the case of the cycle we can achieve even a constant stretch *in expectation*.

1. Show that for any tree T with nonegative lengths of edges that satisfies $\forall i, j : d_G(i,j) \leq d_T(i,j)$, there exist two indices i, j such that $d_T(i,j) \geq (n-1) \cdot d_G(i,j)$.

   Hint 1:

   Gradually turn T into a path wrapping clockwise or anticlockwise around the cycle.

   Each change to T maintains its stretch, while the sum of edge lengths of T drops.

   Hint 2:

   Imagine three vertices $u < v < w$ such that T contains an edge $\{u, w\}$

   of length $w - u$ and an edge $\{v, w\}$ of length $w - v$.

   Change $\{u, w\}$ to $\{u, v\}$ in this particular case.

2. The tree embedding algorithm from the lecture shows that there is a distribution over trees with average stretch $O(\log n)$, i.e., for any i, j we have $\mathbb{E}[d_T(i,j)] \leq O(\log n) \cdot d_G(i,j)$.

   Show that in the case of the cycle there is actually a distribution over trees that achieves the stretch 2.

   Hint:

   Leaving a random edge out of the cycle gives you a path.

2 L1 embedding of cycle [Recommended]

You are given an unweighted cycle C on n vertices.

1. Find a randomized algorithm that embeds the cycle C to $\mathbb{R}$ such that the expected stretch of every edge is constant. That is, your randomized algorithm maps each vertex $u \in C$ to some number $f(u)$. For every pair $u, v \in C$ it has to be the case that $d_C(u,v)/K \leq \mathbb{E}[|f(u) - f(v)|] \leq d_C(u,v)$ for some constant K.

2. Find a deterministic algorithm that embeds C to $\mathbb{R}^2$ with $L1$ norm such that the stretch of every edge is constant. That is, you should map each vertex $u \in C$ to some number $f(u)$. For every pair $u, v \in C$ it has to be the case that $d_C(u,v)/K \leq ||f(u) - f(v)||_1 \leq d_C(u,v)$ for some constant K.

3 Minimum bisection cut

A *bisection cut* is a cut (S, S') such that $|S| = |S'| = n/2$. An *r-balanced cut* is a cut where $r \cdot n \leq |S| \leq (1-r) \cdot n$. A size of a cut is the number of edges that go across the cut.

Give a polynomial-time algorithm that, given a graph G as input, outputs a 1/3-balanced cut whose size is $\mathcal{O}(\log n)$ factor from the size of the smallest-size bisection cut of G.

Hint:

Find a black box reduction to the result you saw in the lecture via a greedy algorithm.