New Hardness Results for Planar Graph Problems in P and an Algorithm for Sparsest Cut

October 8, 2020

Structure of the Talk

- Paper consists of four parts, so today's talk is divided by four parts too.
- I'll try to keep talk parts independent, so if you got lost somewhere, you can join during the next part.

Part 1: Conditional Lower Bounds for Cut Problems in Planar Graphs.

Motivation

- Using fine-grained complexity techniques many (conditional) lower bounds for problems from different areas of theoretical computer science were found.
- Moreover (conditional) lower bounds were found for graph problems restricted to different families of graphs: graphs with bounded diameter, graphs with bounded treewidth, ...
- No (conditional) lower bounds for natural planar graph problems!

Motivation

- Using fine-grained complexity techniques many (conditional) lower bounds for problems from different areas of theoretical computer science were found.
- Moreover (conditional) lower bounds were found for graph problems restricted to different families of graphs: graphs with bounded diameter, graphs with bounded treewidth, ...
- ~~No (conditional) lower bounds for natural planar graph problems!~~
- Today we will see some!

Definitions

- Consider a planar graph $G(V, E)$ with edge costs $c : E \to \mathbb{R}^+$ and vertex weights $w : V \to \mathbb{R}^+$.
- Weight of a set $S \subseteq V$ is just a sum of vertex weights $w(S) = \sum_{v \in S} w(v)$.
- Cut induced by set S is a set of edges (u, v) such that $u \in S, v \notin S$.
- Cost of a cut induced by set S is just a sum of costs of its edges $c(S) = \sum_{(u,v) \in E, u \in S, v \notin S} c(u, v)$.

Problems

- **Sparsest Cut problem**: find a cut minimizing $\frac{c(S)}{w(S) \cdot w(V/S)}$.
- **Minimum Quotient Cut problem**: find a cut minimizing $\frac{c(S)}{\min(w(S), w(V/S))}$.
- **Minimum Bisection problem**: find a cut with $w(S) = \frac{w(V)}{2}$ minimizing $c(S)$.
- **[Park and Philips'93]** Optimal solution is always a cycle in dual graph!
- Uniform version of each of the problems above is just $\forall v \in V : w(v) = 1$

Another Problems

- **The (min, +)-Convolution problem**: you are given two arrays a and b of length n, find an array c such that $c_i = \min_{j=0}^{i} a_j + b_{i-j-1}$.
- **The (min, +)-Convolution Upper Bound problem**: you are given three arrays a, b and c of n integers each, check if $a_i + b_j < c_{i+j}$ for all valid i and j.
- These problems are subquadratic-equivalent and we believe that there is no $O(n^{2-\varepsilon})$ algorithm for $(min, +)-$convolution for $\varepsilon > 0$.
- $(min, +)-$convolution conjecture implies **APSP** conjecture and **3-SUM** conjecture, which imply lower bounds for many different problems.

Lower Bounds

- Today we will show that
  1. If we can solve **Uniform Sparsest Cut problem** in $O(n^{2-\varepsilon})$ time for some $\varepsilon > 0$ in planar graph, we can solve $(min, +)-$convolution problem in $O(n^{2-\delta})$ time for some $\delta > 0$.
  2. If we can solve **Minimum Quotient Cut problem** in $O(n^{2-\varepsilon})$ time for some $\varepsilon > 0$ in planar graph, we can solve $(min, +)-$convolution problem in $O(n^{2-\delta})$ time for some $\delta > 0$.
  3. If we can solve **Minimum Bisection problem** in $O(n^{2-\varepsilon})$ time for some $\varepsilon > 0$ in planar graph, we can solve $(min, +)-$convolution problem in $O(n^{2-\delta})$ time for some $\delta > 0$.
- Surprisingly, we will need only one reduction for all these three problems.

Reduction

Interactive in jamboard.

Part 2: Diameter in Distributed Graphs.

CONGEST Model

- One of the theoretical models in distributed computations.
- Suppose we have a network with n computers and some of them are connected by wires.
- Computers do not know the topology of the network.
- During each round of the communication each computer sends a $\log n$-bit message to each of its neighbors.
- Our goal is to make all the computers know the answer for some problem.
- Complexity is a number of rounds in worst case.

CONGEST Model

- Two important parameters in complexity estimation is n — number of computers and D — diameter of the network graph.
- We say that problem is *tractable* if it's solvable in $poly(D, \log n)$ rounds.
- Many problems were shown to be intractable, for example MST or diameter itself: even deciding whether diameter of a graph is 3 or 4 required $n^{o(1)}$ rounds.
- What about planar networks?

CONGEST Model and Planar Graphs

- Li and Pater showed that diameter is unweighted planar graphs is tractable in CONGEST.
- What about weighted graphs?
- Today we will show that any CONGEST protocol for diameter of weighted planar network with $O(1)$ diameter requires $\Omega(\frac{n}{\log n})$ rounds of communication.

Proof of the Lower Bound

- Lower bound proofs are always hard.
- We will do a reduction from classic two-party communication complexity.

Communication Complexity

- Alice has a private string A of length n, Bob has a private string B of length n.
- They want to evaluate $f(A, B)$.
- Complexity is a number of bits they have to exchange to evaluate $f(A, B)$ in worst case.

String Disjointness

- Alice and Bob have private strings $A, B \in \{0,1\}^n$ and want to check whether for all i $A_i = 0$ or $B_i = 0$ (or both).
- This is a well-studied problem in communication complexity.
- Even with randomness players have to exchange $\Omega(n)$ bits to evaluate their function.
- We will use this lower bound to prove a lower bound for unweighted planar diameter in CONGEST model.

Reduction

Interactive in jamboard.

Part 3: Hierarchical Clustering.

Problem Definition

- We are given n points with some distance function between them.
- Initially we have n clusters with one point in each.
- During the next $n - 1$ rounds two closest clusters are selected and joined into one.

Clusters Distance

- How to define distance between two clusters?
- $MaxDist(A, B) = \max_{a \in A} \max_{b \in B} d(a, b)$
- $MinDist(A, B) = \min_{a \in A} \min_{b \in B} d(a, b)$
- $SumDist(A, B) = \sum_{a \in A} \sum_{b \in B} d(a, b)$

The Problem

- Open question: can we build hierarchical clustering in subquadratic time if closest pair of points can be found in subquadratic time?
- No!
- We will use a distance between vertices in unweighted planar graph as a metric and show that we can't build *MaxDist* and *SumDist* hierarchical clustering under some popular conjecture.
- *MinDist* in easy in that case.

Orthogonal Vectors Conjecture

- We are given n binary vectors $\{0,1\}^d$ with $d = \omega(\log n)$ and have to decide whether there is a pair of orthogonal vectors among them.
- Orthogonal vectors conjecture claims that this problem cannot be solved in $O(n^{2-\varepsilon})$ time for any $\varepsilon > 0$.
- OVC implies lower bounds for many important problems in P.
- We will show that under OVC *MaxDist* and *SumDist* hierarchical clustering in planar graph distance metric cannot be in $O(n^{2-\varepsilon})$ time for any $\varepsilon > 0$.

Proof

- Today we will prove only *MaxDist* case, *SumDist* case is pretty similar.
- We will prove a hardness of a slightly different problem in which initial clusters can be arbitrary.
- It can be shown that for our construction initial state can be achieved during normal hierarchical clustering, but it's quite technical and we have a limited time, so I won't prove it.

Reduction

Interactive in jamboard.

Part 4: New Algorithm for Exact Sparsest Cut.

Park and Phillips algorithm

- We will start from Park and Phillips $O(n^2 W \log C)$ algorithm.
- We will transform our problem to a problem in dual graph, i.e. finding a cycle minimizing $\frac{\text{length}(C)}{\min(\text{inside}(C),\text{outside}(C))}$.
- At first we select and arbitrary spanning tree T of G^* and find its preorder traversal that is consistent with cyclic ordering of vertices around each vertex.
- After that we replace each edge (u, v) in G^* with two directed edges (u, v) and (v, u).
- (u, v) has the same length as original edge in dual graph and weight equals to total weight of vertices enclosed by fundamental cycle induced by (u, v).
- $length(v, u) = length(u, v)$ and $weight(v, u) = -weight(u, v)$.

Park and Phillips algorithm

- Now we are to find a shortest cycle for each total weight from 0 to $\frac{W}{2}$ and select the best.
- This can be done using SSSP-like algorithm.

The Improvement

- We know that there exists $O(\sqrt{n})$-size balance separator in planar graph, which splits a graph into parts of size at most $\frac{2}{3}n$.
- Let's find such separator. If optimal cut passes through it, we just run our SSSP-like algorithm from all of its vertices.
- If optimal cut doesn't pass through separator, just run recursively on halves with a bit modified weights.
- By direct application of a master theorem running time is $O^*(n^{3/2}W)$.