Swiss Winter School on Theoretical Computer Science 2023

Fast Graph Algorithms Using Optimization and Data Structures

R. Kyng *Problem Set 2, v.0 — Tuesday, Jan 31st*

Exercise 1: Expander Basics

Consider an unweighted connected graph $G = (V, E)$. We define the conductance of a cut $S \subseteq V$ in G to be

$$\phi_G(S) = \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}.$$

We define the expansion ϕ_G of G to be the minimum expansion of any non-empty cut. We say that G is a ϕ-expander if $\phi_G \geq \phi$. Notice that $0 < \phi_G \leq 1$.

1. What is the expansion of the complete graph?
2. What is the expansion of the path graph?

Exercise 2: Bounding the Diameter of an Expander

Let G be a connected, unweighted graph that is a ϕ-expander. Prove that the diameter of G is $O(\log m/\phi)$.

Hint: Fix any pair of vertices s, t in G. Then think about running BFS explorations simultaneously from s and t.

Exercise 3: Sparsifying an Expander

Let $G = (V, E)$ be a connected, unweighted graph that is a ϕ-expander with m edges and minimum degree d. Form a new graph $H = (V, E')$ by independently adding each edge of E to E' with probability p.

Show that for some $p = \widetilde{O}(1/d)$, we have that with high probability H is an $\Omega(\phi)$-expander and has at most $\widetilde{O}(m/d)$ edges.

Exercise 4: Routing Recap

The d dimensional hypercube graph $G = (V, E)$ has $V = \{0, 1\}^d$ and has an edge between vertices whose bit strings differ in exactly one coordinate. The graph has $n = 2^d$ vertices and $m = d2^d/2$ edges.

1. The hypercube is a $1/d$-expander. Show that this is tight.
2. Let H be a D-regular graph with 2^d vertices. Construct an embedding of H into G with path lengths bounded by $m^{o(1)}$ and congestion bounded by $m^{o(1)}D$. Note that $d = \widetilde{O}(1)$.

Exercise 5: Dynamic Spanner

In the lectures, we described a decremental spanner for an unweighted graph, as given by the following theorem.

Theorem. *Given a decremental unweighted graph G with n vertices and $m \geq n$ edges and maximum degree D, where G undergoes $t \leq n$ edge deletions and vertex splits, we can maintain a (unweighted, subgraph) spanner H of G with stretch $m^{o(1)}$ for all edges of G, s.t. H undergoes a total of $tm^{o(1)}$ updates (i.e. number of edge insertions and deletions and vertex splits), and the overall running time of the algorithm is $Dm^{1+o(1)}$.*

This theorem falls short of providing a fully dynamic spanner – and it's a little underspecified in terms of how we deal with updates. Let us fix that.

1. We've been fuzzy on the encoding of vertex splits. Suppose that given a vertex v to split into v' and v'', we encode this information via access to the adjacency list $\mathcal{A}_v$ of v and a list $\mathcal{A}_{v'}$ of the edges that move to v', where $\deg(v') \leq \deg(v'')$. Assume the $\mathcal{A}_{v'}$ list includes pointers to the linked list entries in $\mathcal{A}_v$.
   (a) Show that the total encoding size of all updates we receive during the algorithm is $\widetilde{O}(m)$.
   (b) Describe how we can achieve the claimed running time by updating the linked lists in place. You can assume additional information is stored in your data structures, provided you describe how to maintain it efficiently.
2. Using the encoding scheme we just introduced, explain how the encoding size of all updates to H can be bounded by $nm^{o(1)}$.
3. Describe a spanner for an unweighted dynamic graph that in addition to edge deletions and vertex splits also allows edge insertions, while having similar parameters to the theorem above. It's OK to lose some more log factors. Can you make it work even if insertions increase the maximum degree arbitrarily?
4. Extend the spanner to graphs with edge lengths, assuming every edge length is always in the range $[1, m^{10}]$. Again, it's OK to lose some more log factors.

Exercise 6: Spanner Cycle Dichotomy

Consider a graph $G = (V, E)$ with edge lengths $\boldsymbol{l} \in \mathbb{R}_+^E$ and gradients $\boldsymbol{g} \in \mathbb{R}^E$.

Consider the following optimization program:

$$\begin{aligned} \min_{\boldsymbol{\delta} \in \mathbb{R}^{\tilde{E}}} \quad & \boldsymbol{g}^\top \boldsymbol{\delta} \\ \text{s.t. } \boldsymbol{B}\boldsymbol{\delta} &= \boldsymbol{0} \\ \left\| \boldsymbol{L_f} \boldsymbol{\delta} \right\|_1 &\leq 1 \end{aligned} \tag{1}$$

Suppose the optimal value of Program (1) is $-\alpha < 0$ in G.

Let $H = (V, F)$, with $F \subseteq H$ be a spanner of G with an associated collection of spanner cycles

$$\mathcal{C} = \{(e, \text{path for } e \text{ in } H)) \ : e \in E \setminus F\}.$$

Suppose for every $e \in E \setminus F$ the associated spanner cycle has $\boldsymbol{l}(\text{path for } e \text{ in } H) \leq \gamma \boldsymbol{l}(e)$, some some $\gamma \geq 1$.

Prove that if every spanner cycle satifies

$$\frac{\boldsymbol{g}(e) + \sum_{h \in \text{path for } e \text{ in } H} \boldsymbol{g}(h)}{\boldsymbol{l}(e) + \sum_{h \in \text{path for } e \text{ in } H} \boldsymbol{l}(h)} \geq -\frac{\alpha}{2(1+\gamma)}$$

then the value of Program (1) restricted to the edges of H has value $\leq -\frac{\alpha}{2\gamma}$.

Bonus Exercise: Expander Decomposition

by Maximilian Probst Gutenberg.

Let $G = (V, E)$ be a connected, undirected graph. In this problem, you will show that there is an algorithm that computes a ϕ-expander decomposition $X_1, X_2, \ldots, X_k$ for G of with $\tilde{O}(\phi m)$ edges crossing between the partitions in time $O(m \log^c n)$ for some constant c.

Assume that you are given an algorithm $\textsc{CertifyOrCut}(G, \phi)$ that given a graph G and a parameter ϕ either:

- *Certifies* that G is a ϕ-expander, or
- Presents a *cut* S such that $\phi(S) = \tilde{O}(\phi)$.

The algorithm $\textsc{CertifyOrCut}(G, \phi)$ runs in time $O(m \log^{c'} n)$ for $c' > 0$.

1. Show that there is an algorithm that uses $\textsc{CertifyOrCut}(G, \phi)$ and computes a ϕ-expander decomposition with $\tilde{O}(\phi m)$ edges crossing in time $O(mn \cdot \log^{c'} n)$.

2. Show that in $O(mn \cdot \log^{c'} n)$ time, you can implement a procedure $\textsc{CertifyOrLargeCut}(G, \phi)$ that outputs a set S (possibly empty) with $\phi(S) = \tilde{O}(\phi)$ such that either

   - $G[V \setminus S]$ is a ϕ-expander and $vol_G(V \setminus S) \geq \frac{1}{3}m$, or
   - $\min\{vol_G(S), vol_G(V \setminus S)\} \geq \frac{1}{3}m$.

   Hint: Prove that given a set $S \subseteq V$ of conductance $\phi_G(S) \leq \phi$ and a set $S' \subseteq V \setminus S$ in $G[V \setminus S]$ with conductance $\phi_{G[V \setminus S]}(S') \leq \phi$, you have that $\phi_G(S \cup S') \leq \phi$.

3. Assume that there is an algorithm that implements $\textsc{CertifyOrLargeCut}(G, \phi)$ to run in time in $O(m \log^{c''} n)$ time[1]. Show that this implies an $O(m \log^{c''+1} n)$ time algorithm to compute a ϕ-expander decomposition.

[1]This implementation can be derived rather straight-forwardly from the algorithm in "Graph partitioning using single commodity flows." by Khandekar, Rohit, Satish Rao, and Umesh Vazirani which appeared in the Journal of the ACM (JACM) 56.4 (2009).