COS 597A: Recent Advances in Graph Algorithms (Fall '24) September 16, 2024

Lecture 4: The Expander Hierarchy and its Applications

Instructor: *Huacheng Yu* Scribe notes by: *Elena Gribelyuk*

Introduction

In this lecture, we begin our study of the recent paper on maintaining the expander hierarchy of a graph under dynamic updates [1]. Given an undirected graph $G = (V, E)$ which may undergo arbitrary (potentially adaptive) edge insertions or deletions, our goal is to design a data structure that maintains the expander hierarchy of G with fast total update time. Today, we focus on defining the expander hierarchy of a graph, as well as providing an initial overview of the algorithm in an important special case where all updates to the graph arrive as a *batch*. Before we formally define "expander hierarchy," we recall some important background from the previous lecture.

Notation. For an undirected, unweighted graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, we define the *set of edges in cut* (S, T) by $E(S, T) = \{(u, v) : u \in S, v \in T\}$. Additionally, for any subset of vertices $S \subseteq V$, we define the *volume* of S by $\text{vol}(S) = \sum_{u \in S} \deg(u)$. As before, we denote $\tilde{O}(f) = O(f \cdot \text{polylog}(n))$.

Now, we recall the following definition of a ϕ-expander:

Definition 1 (ϕ-expander). *A graph $G = (V, E)$ is a ϕ-expander if, for any set $S \subset V$ such that $S \neq \emptyset$ and $S \neq V$, we have that*

$$|E(S, V \setminus S)| \geq \phi \cdot \min\left(vol(S), vol(V \setminus S)\right)$$

We also define a generalized notion of an induced subgraph, which will be crucial to the stronger notion of "boundary-linked" expander decomposition which we introduce in the next section.

Definition 2. *Let $G[S]$ be the subgraph induced by vertex set $S \subseteq V$. Then, for any integer $w \geq 1$, we let $G[S]^w$ denote the graph obtained after the following transformation is applied to $G[S]$: for each $v \in S$, add w self-loops for each boundary edge $(v, x) \in E$ where $x \notin S$.*

Notably, we observe that $G[S]^0 = G[S]$, and vertices in $G[S]^1$ have the same degree as in the original graph G.

Boundary-linked Expander Decomposition

We recall the traditional notion of a ϕ-expander decomposition of a graph G, which has found numerous applications in the graph algorithms literature. This definition was covered in the previous lecture, but we restate it here for comparison with our stronger notion of "boundary-linked" expander decomposition, which we will focus on next.

Definition 3 (ϕ-expander decomposition). *For any graph $G = (V, E)$ and any $\phi > 0$, there is a partition $\mathcal{U} = \{U_1, ..., U_k\}$ of V into clusters, such that*

1. *$\sum_{i\in[k]} |E(U_i, V \setminus U_i)| = \tilde{\mathcal{O}}(\phi \cdot m)$*

2. *For all $i \in [k]$, $G[U_i]$ is a ϕ-expander.*

Intuitively, this definition states that we can always find a partition of vertices $(U_1, ..., U_k)$ such that, after we remove a $\tilde{\mathcal{O}}(\phi)$-fraction of the edges from G, the remaining induced subgraphs $G[U_i]$ are all ϕ-expanders.

A Stronger Expander Decomposition. The central contribution of this work is a stronger notion of expander decomposition, which the authors call "boundary-linked" expander decomposition. Compared to the previous definition, this decomposition has the property that the vertices inside each U_i are better connected to each other than to vertices in $V \setminus U_i$. Here, we focus on the setting where $\alpha = \frac{1}{\text{polylog}(n)} = \tilde{\Omega}(1)$ and $\phi = \frac{1}{n^{O(1)}} << \alpha$.

Definition 4 ((α, ϕ)-boundary linked expander decomposition of G)**.** *For any $G = (V, E)$, a (α, ϕ)-boundary linked expander decomposition of G is a partition $\mathcal{U} = \{U_1, ..., U_k\}$ of vertices with $\phi_1, ..., \phi_k \geq \phi$ such that*

1. *$\sum_{i=1}^{k} |E(U_i, V \setminus U_i)| \leq \tilde{\mathcal{O}}(\phi \cdot m)$.*

2. *For all $i \in [k]$, $G[U_i]^{\alpha/\phi_i}$ is a ϕ_i-expander.*

3. *For all $i \in [k]$, $|E(U_i, V \setminus U_i)| \leq \tilde{\mathcal{O}}(\phi_i \cdot vol_G(U_i))$.*

Importantly, this notion of expander decomposition is *stronger* than the previous definition, as we require the decomposition $\mathcal{U}$ to satisfy a *strengthened expander property* for each cluster U_i (property 2), as well as requiring each cluster U_i to be *more connected to itself than to other vertices* (property 3). In fact, these are actually very intuitive properties of a "good clustering" of vertices in G:

- First, we observe that by adding $w = \alpha/\phi_i$ self-loops to boundary vertices of $G[U_i]$, we effectively increase the volume of each boundary vertex v, making the ϕ-expander property *harder* to satisfy. Specifically, this stronger property 2 implies that for each vertex $v \in U_i$, we have that

$$\deg_{G[U_i]}(v) \geq \alpha \cdot \deg_{G[V \setminus U_i]}(v).$$

  In particular, if we did not enforce this property, it could be that $\deg_{G[U_i]} << \deg_{G[V \setminus U_i]}$, which does not align well with our intuition of a cluster.

- Secondly, property 3 ensures that for most vertices $v \in U_i$, we have that

$$\deg_{G[V \setminus U_i]}(v) \leq \tilde{\mathcal{O}}(\phi_i) \cdot \deg_{G[U_i]}(v).$$

  Again, this is aligned with our intuition of a cluster, in which vertices should be well-connected with each other and have few connections with other clusters in G.

Construction. With this definition in mind, note that we can iteratively construct a (α, ϕ)-boundary-linked expander decomposition as follows: as long as property 2 is not satisfied for some cluster U_i, we find the sparsest cut in U_i and separate U_i into two parts. Using a similar argument as in the construction of ϕ-expander decomposition, it can be shown that after this iterative process terminates, the resulting decomposition satisfies properties 1-3 of the definition above. In fact, we can show the following theorem.

Theorem 5. *A (α, ϕ)-boundary-linked expander decomposition can be computed in $\tilde{\mathcal{O}}\left(mn^{O(1)}\right)$ time, for $\alpha = \frac{1}{polylog(n)}$ and $\phi = \frac{1}{n^{O(1)}}$.*

Main Result: Maintaining the Expander Hierarchy

Expander Hierarchy. In this section, we formally define the *expander hierarchy* of a graph. For input graph $G_0 = G$ and (α, ϕ)-boundary-lined expander decomposition $\mathcal{U}_0 = (U_1^{(0)}, ..., U_k^{(0)})$ of G, we obtain the next graph G_1 of the expander hierarchy by **contracting each cluster U_i into 1 vertex** (when doing this, we remove self-loops and keep parallel edges). This step produces contracted graph G_1, and we then continue this process by finding the (α, ϕ)-boundary-linked expander decomposition $\mathcal{U}_1 = (U_1^{(1)}, ..., U_\ell^{(1)})$ of G_1 and similarly contracting each cluster $U_i^{(1)}$ into just one vertex. We continue to iteratively find the (α, ϕ)-boundary-linked expander decomposition and contract the current graph G_i as described above, until we reach the step when $G_k = (V_k, E_k)$ has only one vertex and $|E_k| = 0$.

Definition 6 (Expander Hierarchy of a Graph)**.** *The sequence of graphs $\{G_0, ..., G_k\}$ generated through the process above is the expander hierarchy of G.*

Note that by property 1, we have $|E_{i+1}| \leq \tilde{\mathcal{O}}\left(\phi \cdot |E_i|\right)$, so it follows that $k \leq \mathcal{O}(\log_{1/\phi}(m)) \leq \mathcal{O}(\log^{1/4}(n))$.

Main result. At this point, we are ready to state the main result of [1].

Theorem 7. *For any graph G, the expander hierarchy of G can be maintained in time $n^{O(1)}$ per edge insertion or deletion.*

The dynamic algorithm for maintaining the expander hierarchy of a graph implies efficient algorithms for many important graph problems. For instance, given the dynamic algorithm of [1], we can maintain the connected components of each G_i in the expander hierarchy. Additionally, the expander hierarchy can be used to answer approximate (s, t)-min-cut queries.

Update time & Recourse. Recall that the expander hierarchy of G is given by a sequence of graphs $(G_0, ..., G_k)$, where G_{i+1} is generated from G_i by contracting each cluster $U_j^{(i)}$ in the (α, ϕ)-boundary-linked expander decomposition into a single vertex. So, it suffices for us to design an algorithm to maintain an (α, ϕ) decomposition $\mathcal{U}$ and $G_{\mathcal{U}}$ (also known as G_1). For this, there are two important measures that we should focus on while designing the algorithm:

- **Update time**: the time for computing the updated $\mathcal{U}$ and $G_{\mathcal{U}}$.
- **Recourse**: the number of edges that need to be updated in $G_{\mathcal{U}}$. We denote the recourse by ρ.

Since we will eventually want to maintain a k-level expander hierarchy, the number of edge updates to the top level will be $\leq \rho^k$. Thus, we would like to minimize the recourse for each G_i. With this in mind, the authors show the following theorem.

Theorem 8. *For $\alpha = \frac{1}{polylog(n)}$ and $\phi = 2^{-\log^{3/4}(n)}$, a (α, ϕ)-boundary-linked expander decomposition of G can be maintained with*

1. *Update time $2^{\tilde{\mathcal{O}}\left(\log^{\frac{3}{4}}(n)\right)}$*
2. *Recourse $\rho \leq 2^{\tilde{\mathcal{O}}\left(\sqrt{\log(n)}\right)}$*

In particular, this implies that the total number of updates to the top level of the expander hierarchy

$$\rho^k \leq \left(2^{\mathcal{O}\left(\sqrt{\log(n)}\right)}\right)^{\log^{1/4}(n)} = 2^{\mathcal{O}\left(\log^{3/4}(n)\right)}$$

for $k = \mathcal{O}(\log_{1/\phi}(m)) = \mathcal{O}(\log^{1/4}(n))$. Thus, the total time per update is $2^{\mathcal{O}\left(\log^{3/4}(n)\right)}$.

Overview of the algorithm

As an introduction to the algorithm, we will discuss how the algorithm handles an important special case: suppose that all updates to G arrive as *one batch*, and the task is to efficiently maintain the expander hierarchy after processing these updates. Moreover, for now we will focus on minimizing recourse.

Consider the (α, ϕ)-boundary-linked expander decomposition $\mathcal{U} = (U_1, ..., U_k)$ of G (denoted $ED(G)$ below), and let us focus on one cluster U_1. Suppose there are k updates with at least 1 endpoint incident to U_1. We consider the following two possible cases:

1. If $k \geq \tilde{\Omega}(\text{vol}_G(U_1) \cdot \phi_1)$, then we will simply rebuild, or "reset," the cluster U_1 by building a (α, ϕ_i)-boundary-linked expander decomposition on $G[U_1]^{\alpha/\phi_i}$. In this case, note that the recourse can be computed as follows:

$$\begin{aligned}\rho &= \#(\text{outgoing edges from } U_1) + \#(\text{edges between components of } ED(U_1)) \\ &\leq \tilde{\mathcal{O}}\left(\phi_1 \cdot \text{vol}(U_1)\right) + \phi_1 \cdot \text{vol}(G[U_1]^{\alpha/\phi_1}) \leq \phi_1 \left(\text{vol}(G[U_1]) + \frac{\alpha}{\phi_1} \cdot \tilde{\mathcal{O}}\left(\phi_1 \text{vol}(U_1)\right)\right) \leq \tilde{\mathcal{O}}\left(\phi_1 \cdot \text{vol}(U_1)\right)\end{aligned}$$

   So, it follows that the recourse is $\tilde{\mathcal{O}}(1)$ per update.

2. Alternatively, suppose that $k << \text{vol}_G(U_1) \cdot \phi_1$. In this case, the authors show that there always exists a subset of vertices $P \subseteq U_1$ satisfying the following properties:
   (a) $\text{vol}(P) \leq \tilde{\mathcal{O}}(k/\phi_1)$
   (b) $E(P, U \setminus P) \leq \tilde{\mathcal{O}}(k)$
   (c) $G[U_1 \setminus P]^{\alpha/\phi_1}$ is a $\Omega(\phi_1)$-expander.

Then, we consider the graph after pruning this set P: we replace U_1 by $U_1 \setminus P$ and build a (α, ϕ_1)-boundary-linked expander decomposition $\mathcal{U}' = (U'_1, ..., U'_{k'})$ of $G[P]^{\alpha/\phi_1}$ and replace U_1 in $\mathcal{U}$ to get a new partition $(U_1 \setminus P, U'_1, ..., U'_{k'})$ of U_1. By definition, this new partition is an (α, ϕ_1)-boundary-linked expander decomposition of G. To bound the recourse, we compute as follows:

$$\rho \leq |E(P, U_1 \setminus P)| + |E(P, V \setminus U_1)| + \#(\text{edges from expander decomposition of } P)$$

Note that (a) $= |E(P, U_1 \backslash P)| \leq \tilde{\mathcal{O}}(k)$ by definition of P, and (b) $= |E(P, V \backslash P)| \leq \tilde{\mathcal{O}}(|E(P, U_1 \setminus P)|) \leq \tilde{\mathcal{O}}(k)$ as well. Additionally, we have that $\#(\text{edges from ED of } P) \leq \tilde{\mathcal{O}}\left(\phi_1 \cdot \text{vol}(G[P]^{\alpha/\phi_1}\right) \leq \phi_1 \cdot \left(\text{vol}(P) + \frac{\alpha}{\phi_1}((\text{a}) + (\text{b})\right) \leq \tilde{\mathcal{O}}(k)$. Consequently, we can see that

$$\rho \leq \tilde{\mathcal{O}}(k).$$

As before, this corresponds to recourse $\tilde{\mathcal{O}}(1)$ per update.

With these two cases in mind, the main idea of the algorithm is as follows: we apply cases 1-2 to each cluster U_i separately, but if the total number of updates is larger than $\text{vol}(G) \cdot \phi$, then we rebuild the entire decomposition of G.

References

[1] Gramoz Goranci et al. "The expander hierarchy and its applications to dynamic graph algorithms". In: *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '21. Virtual Event, Virginia: Society for Industrial and Applied Mathematics, 2021, pp. 2212–2228. ISBN: 9781611976465.