University of Michigan–Ann Arbor

Department of Electrical Engineering and Computer Science

EECS 498 004 **Advanced Graph Algorithms**, Fall 2021

Lecture 28: Summary and Beyond

December 9, 2021

Instructor: Thatchaphol Saranurak

Scribe: Gary Hoppenworth

1 What have we learned?

1.1 5 Equivalent Views of Expanders (w.r.t Conductance)

1. **Robust against deletions.** We have seen that expanders are robust against deletions in the sense that when you remove a 'small' number of edges from an expander, it will only disconnect a 'small' volume of edges from the graph. More formally, let G be a ϕ-expander undergoing d insertions/deletions. Then there is a $P \subset V(G)$ such that P has small volume ($\mathrm{vol}(P) \leq O(d/\phi)$) and $G \setminus P$ is a $\Omega(\phi)$-expander.

2. **Cut view.** Expanders are well-connected graphs with respect to cuts. Formally, if graphs G and H share the same vertex set, and every vertex in H has smaller degree than the corresponding vertex in G, then we say that $H \preccurlyeq^{\mathrm{deg}} G$. Likewise, we say that $H \preccurlyeq^{\mathrm{cut}} G$ if every cut $(S, \overline{S})$ in H has fewer edges than the corresponding cut in G. We have seen that

$$G \text{ is } \phi\text{-expander and } H \preccurlyeq^{\mathrm{deg}} G \implies H \preccurlyeq^{\mathrm{cut}} \frac{1}{\phi} G$$

   So G is the most well-connected graph (within a $1/\phi$ factor) with respect to cuts among all graphs with the same vertex degree profile.

3. **Flow view.** Expanders are well-connected graphs with respect to flow. We say that $H \preccurlyeq^{\mathrm{flow}} G$ if there is a flow in G that can route the demand H with no congestion. The problem of determining whether $H \preccurlyeq^{\mathrm{flow}} G$ is called the maximum concurrent flow problem. We saw a generalization of the max-flow min-cut theorem, which stated that

$$H \preccurlyeq^{\mathrm{flow}} G \implies H \preccurlyeq^{\mathrm{cut}} G \implies H \preccurlyeq^{\mathrm{flow}} O(\log n) G$$

   It immediately follows from the cut view of expanders that ϕ-expanders are the most well-connected graphs with respect to flow (within a $O(\log n/\phi)$ factor) among all graphs with the same vertex degree profile.

4. **Eigenvalue view.** Given a graph G, we saw how to define the Laplacian $\boldsymbol{L}_G$ and normalized Laplacian $\boldsymbol{N}_G$ matrices of G. We saw that the second eigenvalues of these matrices are closely related to the conductance $\Phi(G)$ of G. We proved Cheeger's inequality which states that

$$\frac{\lambda_2(\boldsymbol{N}_G)}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2(\boldsymbol{N}_G)}.$$

   Because ϕ-expanders are the graphs with $\Phi(G) \geq \phi$, this gives another view of expanders from a spectral perspective.

5. **Random walks.** We defined a lazy random walk on graphs and showed that no matter where you start in the graph, the probability distribution of your location in the graph at time t will eventually converge to a stationary distribution. In this stationary distribution, the probability you are at specific vertex at time t is proportional to the degree of that vertex. Given a graph G, we can define the mixing time, $\tau_{\mathrm{mix}}(G, \epsilon)$, as the number of steps it takes a random walk in G to 'mix' with G so that the probability distribution of its location will be close to the stationary distribution within an ϵ amount. We saw that

$$\tau_{\mathrm{mix}}(G, \epsilon) \approx 1/\lambda_2(\boldsymbol{N}_G)$$

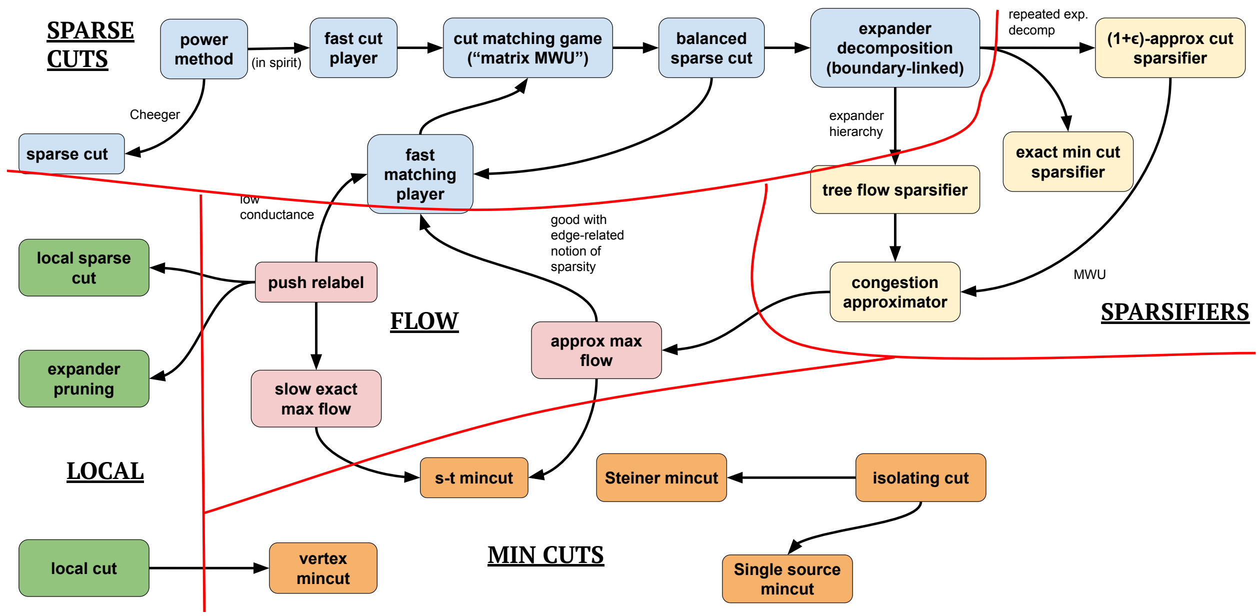
   We know good expanders have large $\lambda_2(\boldsymbol{N}_G)$ from Cheeger's inequality. This tells us random walks on expanders will have a small mixing time.

1.2 Graph Algorithm Toolkit / Concepts

We saw many algorithmic tools and concepts that are useful for solving problems related to graphs and expanders. Here are some of them.

- Metric embedding
- Power method
- Isolating cuts
- Local cuts
- Cut-matching games
- Balanced cuts
- Expander decomposition and its variants
- Sparsifiers in various forms
- Push relabel
- MWU
- Approximate max flows

These ideas are all connected.



2 More Problems that can be solved using Related Techniques

We have learned many important techniques from this class, and we have seen many state-of-the-art algorithms. Below, I included additional important problems that we haven't learn about, but material from our class will help you understand/improve the state-of-the-art.

2.1 Minimum cuts

1. Global minimum cuts in $\tilde{O}(m)$ randomized time[1]
   - In class: max flow time (isolating cuts), $\tilde{O}(mk^2)$ time (local cuts)
   - Ingredient: Graph sparsification, MWU
2. Global minimum cuts in $m^{1+o(1)}$ deterministic time[2]
   - Ingredient: Deterministic graph sparsification (Expander hierarchy, Pessimistic estimator method for derandomization)
3. Vertex connectivity in max flow time[3]
   - In class: $\tilde{O}(mk^2)$ time (local cuts)
   - Ingredient: Isolating cuts, graph sketching
   - Exciting open questions:
     - $\tilde{O}(m)$ time (even for $(1+\epsilon)$-approximation)
     - $o(mn)$ for weighted case

[1] https://arxiv.org/pdf/cs/9812007.pdf
[2] https://arxiv.org/abs/2106.05513
[3] https://arxiv.org/pdf/2104.00104.pdf

2.2 Sparse cuts

1. Expander decomposition in $\tilde{O}(m/\phi)$ time[4]
   - In class: $m^{1+o(1)}$ time
   - Ingredient: *non-stop* version of cut-matching game
2. Deterministic expander decomposition in $m^{1+o(1)}$ time
   (a) Ingredient: multi-commodity cut-matching game, dynamic shortest path
   (b) Exciting open problem: $\tilde{O}(m)$ time
3. Expander decomposition in distributed networks in polylog(n/ϕ) rounds[5]
   - Ingredient: Basics on the distributed computation model, *non-stop* cut-matching game
   - Many applications in distributed computations: finding cliques in distributed network and so on
4. Local sparse cuts starting from a single node inside a sparse cut.
   - In class: given a set A with a sparse cut S where overlapping a lot with A (i.e. $\mathrm{vol}(S \cap A) \geq \sigma \mathrm{vol}(S)$). Time: $O(\frac{\mathrm{vol}(A)}{\phi\sigma})$
   - Ingredient: Local PageRank[6], Local random walk[7]

2.3 Graph sparsification

1. Deterministic $(1+\epsilon)$-approximate cut/spectral sparsifier in poly(n) time
   - In class: randomized (we only describe the algorithm, no proof)
   - Ingredient: more spectral graph theory
2. Vertex sparsifier for cuts with quality $O(\frac{\log|T|}{\log\log|T|})$ in poly(n) time [8]
   - Problem: Given a graph $G = (V, E)$ and a terminal T, find a graph H where $V(H)=T$ and, for all $A, B \subseteq T$, we have

     $$\mathrm{mincut}_G(A,B) \leq \mathrm{mincut}_H(A,B) \leq O(\frac{\log|T|}{\log\log|T|}) \cdot \mathrm{mincut}_G(A,B).$$

   - In class: using boundary-linked decomposition: $O(\log n)$ quality, $V(H) \supset T$, and only for unweighted graphs
3. Vertex sparsifier for cut of size at most c in time $\tilde{O}(mc^{O(c)})$[9] or poly(n)[10]

[4]https://arxiv.org/pdf/2007.14898.pdf
[5]https://arxiv.org/pdf/2007.14898.pdf
[6]http://www.math.ucsd.edu/~fan/wp/localpartition.pdf
[7]Chapter 22 of http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf
[8]https://epubs.siam.org/doi/pdf/10.1137/130908440
[9]https://arxiv.org/abs/2007.07862
[10]https://arxiv.org/abs/2011.15101

- Problem: Given a graph $G = (V, E)$ and a terminal T, find a graph H where $V(H) \supseteq T$ and, for all $A, B \subseteq T$, we have

$$\min\{c, \mathrm{mincut}_G(A, B)\} = \min\{c, \mathrm{mincut}_H(A, B)\}.$$

- In class: not exact (boundary-linked decomposition)
- Ingredient: boundary-linked decomposition, representative sets based on matroid

4. Tree flow sparsifier with quality polylog(n)[11]

   - In class: quality $n^{o(1)}$ via expander hierarchy
   - Ingredient: *non-stop* cut-matching game

2.4 Flow

Let $G = (V, E)$ be an undirected graph and let $\mathsf{d} \in \mathbb{R}^V$ be a demand.

In the max flow problem (a.k.a. ℓ_∞-flow problem), find a flow f satisfying d such that $\max_{e \in E} |f(e)|$ is minimized.

1. (ℓ_2-flow): Electrical flow and Laplacian solvers in $\tilde{O}(m)$ time

   - Problem: find a flow f satisfying d such that $\sum_{e \in E} f(e)^2$ is minimized
     - Equivalent problem: find a potential vector x where $\boldsymbol{L}_G x = \mathsf{d}$.
   - Ingredient: Graph sparsification, Conjugate gradient descent (many other interesting ways)

2. (ℓ_1-flow): Transshipment and shortest paths in $\tilde{O}(m)$ work and $\tilde{O}(1)$ depth (parallel time)[12]

   - Problem: find a flow f satisfying d such that $\sum_{e \in E} |f(e)|$ is minimized
   - Ingredient: MWU, metric embedding

3. Multi-commodity flow on expanders in $m^{1+o(1)}$ time.

   - Ingredient: dynamic shortest paths[13] or random-walk based algorithm[14]
   - Exciting open problem: $\tilde{O}(m)$ time

4. Exact max flow in $\tilde{O}(m^{1.5-\epsilon})$

   - Ingredient: dynamic spectral sparsification, interior point method
   - Exciting open problem: $\tilde{O}(m)$ time

[11] https://www.youtube.com/watch?v=XAdM0CqrxOI https://epubs.siam.org/doi/pdf/10.1137/1.9781611973402.17

[12] https://arxiv.org/pdf/1911.01626.pdf

[13] Section 3.2 in https://arxiv.org/pdf/2009.08479.pdf

[14] https://groups.csail.mit.edu/tds/papers/Ghaffari/podc117.pdf

2.5 Other Problems

There are other problems not so related to techniques in this class (at least not yet). But I want to mention them anyway.

1. All-pair shortest paths in $O(n^3/2^{O(\sqrt{\log n})})$ time.
   - Exciting question: $O(n^{2.99})$ time.
2. Parallel s-t reachability in $m^{1+o(1)}$ work and $n^{0.5+o(1)}$ depth
   - Exciting question: $\tilde{O}(m)$ work and polylog(n) depth.

3 Related Fields

There are many other exciting questions beyond this. For example, we can try to make everything taught in this class...

- dynamic (my field)
- distributed/parallel
- sub-linear space/sub-linear time.

4 Final words

- I hope you learn a lot and push the state-of-the-art further.
- Thank you for your hard work throughout the course.