| | |
|---|---|
| **CS 598CSC: Topics in Graph Algorithms** | **Lecture date: 09/19/2024** |
| **Instructor: Chandra Chekuri** | **Scribe: CC** |

# 1 Well-linked Sets and connections to Expansion and Treewidth

Recall that an $\alpha$-expander for some parameter $\alpha > 0$ is a graph $G = (V, E)$ such that for all sets $S \subset V$ with $|S| \leq |V|/2$, $|\delta(S)| \geq \alpha|S|$. This is a cut-condition. What does this imply? Suppose $A, B$ are two disjoint sets of vertices of equal size, that is $|A| = |B|$. Clearly $|A|, |B| \leq |V|/2$.

**Definition 1.** *Let $A, B$ be two disjoint sets of equal seize. An $A$-$B$ linkage is a set of $|A|$ edge-disjoint paths such that each vertex in $A \cup B$ is the end-point of exactly one of the paths. We say that $A$ and $B$ are linked in $G$ if they there is an $A$-$B$ linkage. We say that an $A$-$B$ linkage is fractional if there is a flow in $G$ with that satisfies demand of 1 on each vertex in $A$ and a demand of $-1$ on each vertex of $B$ (note that this is single-commodity flow). We say that $A, B$ are $\alpha$-linked for some parameter $\alpha$ if there is a flow in $G$ that satisfies the demand of $\alpha$ on each end point in $A$ and a demand of $-\alpha$ on each vertex of $B$.*

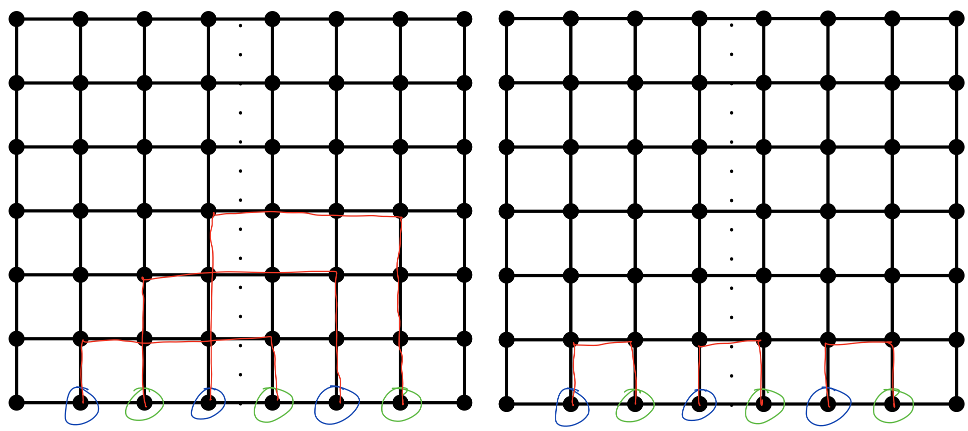

Figure 1: $A, B$ linkage in a grid where $A$ is shown in blue and $B$ in green. The paths are shown in red. On the left the paths are edge-disjoint and on the right they are node-disjoint.

We claim the following.

**Claim 1.** *Suppose $G$ is an $\alpha$-expander with $\alpha \geq 1$. Then there is an $A$-$B$ linkage in $G$ for every pair of disjoint equal sized sets $A, B$.*

*Proof.* To check if there are the desired edge-disjoint paths, we can create a flow problem by adding two new vertices $s, t$, connecting $s$ to vertices in $A$ with an edges of capacity 1, and connecting vertices in $B$ to $t$ with edges of capacity 1, and finding an $s$-$t$ maxflow. Let $H = (V \cup \{s, t\}, E_H)$ be this new graph. If there is an $s$-$t$ flow of value $k$ in $H$ then these correspond to the desired paths (why?). Suppose the flow is strictly less than $k$. Then by maxflow-mincut theorem there is a set $S' \subseteq V_H$ with $s \in S, t \notin S''$ such that $|\delta_H(S')| < k$. Let $S - s$ be the set of vertices in $G$. It is not difficult to see that $|\delta_H(S')| = |\delta_G(S)| + |A \cap (V - S)| + |B \cap S|$. This implies that $S \neq \emptyset$ for if it were then $|A \cap (V - S)| = k$. Similarly $V - S \neq \emptyset$. Suppose $|S| \leq |V - S|$ (the smaller side). Then by expansion guarantee, $|\delta_G(S)| \geq |S|$ which implies that $|\delta_G(S)| \geq |A \cap S|$ but then $|\delta_H(S')| = |\delta_G(S)| + |A \cap (V - S)| + |B \cap S| \geq |A \cap S| + |A \cap (V - S)| \geq k$ contradiction our assumption on the $s$-$t$ mincut. Similar reasoning applies if $|V - S| \leq |S|$. ■

One can scale capacities or directly prove the following corollary.

**Corollary 2.** *Suppose $G = (V, E)$ is an $\alpha$-expander. Then if $A, B$ are disjoint vertex sets with $|A| = |B|$ then $A, B$ are $\alpha$-linked.*

Interestingly the converse is also true.

**Claim 3.** *Suppose $G = (V, E)$ is a graph and for* any *two disjoint sets $A, B$ of equal size, $A, B$ are $\alpha$-linked. Then $G$ is an $\alpha$-expander.*

*Proof.* Let $S \subset V$ with $|S| \leq |V|/2$. Let $T \subseteq V - S$ be an arbitrary set with $|T| = |S|$, then $|S| = |T|$ and $S, T$ are disjoint. If $S, T$ are $\alpha$-linked then we cannot have $|\delta_G(S)| < \alpha|S|$ due to the maxflow-mincut theorem. ■

**Remark 4.** *We do not have to insists on $A, B$ being disjoint. If they are not disjoint and allow each vertex in $A \cap B$ to connect to itself via an empty path then it is the same as asking $A - B$ and $B - A$ to be linked. Thus, in some settings the definitions only require $A, B$ to be of same size. We kept the disjointness explicit to make it easier to understand and visualize.*

**Definition 2.** *Let $G = (V, E)$ be a graph. A set $X$ is $\alpha$-well-linked in $G$ (we use well-linked if $\alpha = 1$) if for any two $A, B \subseteq X$ with $|A| = |B|$, the sets $A, B$ are $\alpha$-linked.*

The following lemma can be shown in a similar fashon as the previous ones and we leave its formal proof as an expander.

**Lemma 5.** *A set $X \subseteq V$ is $\alpha$-well-linked in $G$ iff for any set $S \subset V$, $|\delta_G(S)| \geq \alpha \min(|S \cap X|, |S \cap (V - X)|)$.*

**Corollary 6.** *A graph $G = (V, E)$ is an $\alpha$-expander iff $V$ is $\alpha$-well-linked in $G$.*

Thus, the notion of well-linked sets extends the definition of expansion to *subsets* of the graph and this is very useful in a number of settings. For instance a $\sqrt{n} \times \sqrt{n}$ grid is a planar graph with $n$ vertices. It has a bisection with $O(\sqrt{n})$ edges hence it is at best a $1/\sqrt{n}$-expander (which in fact it is). It has a well-linked set of size $\Omega(\sqrt{n})$.

**Remark 7.** *A star on $n$ vertices is an expander and has a well-linked set of size $|V|$ which may seem strange but this is an artificat of the fact that the degree of the center vertex is very large. This artifact disappears if we ask for constant degree graphs or if we insist on node well-linkedness which we do in the discussion on treewidth.*

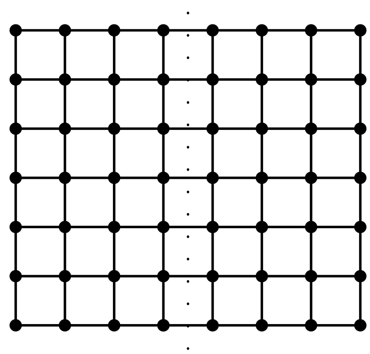

Figure 2: A $\sqrt{n} \times \sqrt{n}$ grid has a well-linkset set of size $\sqrt{n}$. Take the vertices on the bottom row and verify that it is well-linked.

### 1.1 Weighted well-linked sets and conductance

We can define well-linkedness with respect to weights on the vertices. Recall that had considered product multicommodity flows during the discussion on sparsest cut. We can define weighted notion of well-linked sets. Instead of using linkages which is clunky in this setting, we use the cut definition.

**Definition 3.** *Let $\pi : X \to \mathbb{Q}_+$ be non-negative weights on $X \subseteq V$. We say that $X$ is $\pi$-weighted well-linked in in $G$ if for any $S \subset V$, $|\delta_G(S)| \geq \min(\pi(S \cap X), \pi(S \cap (V - X)))$.*

**Remark 8.** *Typically it is assumed that $\pi(v) \leq c\pi(V)$ for some constant $c < 1$ (say $c = 1/3$). That is, no single vertex is too heavy compared to the total weight.*

Note that if $\pi(v) = 1$ for all $v \in X$ we get back the definition of well-linkedness. If $\pi(v) = \deg(v)$ for each $v \in V$ we obtain the definition of conductance. What does this weighted definition mean in terms of linkages? One useful way to think about $\pi$ being integer valued. In this case we replace each vertex with weight $\pi(v)$ with a clump of $\pi(v)$ vertices connected to $v$ (say in a star like fashion) and use these as new vertices $Y$. Essentially $X$ being $\pi$-weighted well-linked is the same as $Y$ being well-linked.

### 1.2 Node capacitated linkages and treewidth

We can generalize the notion of linkages to require node-disjoint paths instead of edge-disjoint paths. In graph theory literature on treewidth, the notion of linkages is defined primarily via node-disjoint paths. We will not use treewidth very much in this course and hence we overload edge and node well-linked notations. We also skip the definition of $\alpha$-linkage for now. The definition is basically the same where we want flow with node capacities rather than edge capacities.

**Definition 4.** *Let $A, B$ be two disjoint sets of equal seize. An $A$-$B$ linkage is a set of node-disjoint paths $|A|$ paths such that each vertex in $A \cup B$ is the end-point of exactly one of the paths. We say that $A$ and $B$ are linked in $G$ if they there is an $A, B$ linkage.*

The notion of well-linked sets naturally generalizes.

**Definition 5.** *Let $G = (V, E)$ be a graph. A set $X$ is $\alpha$-well-linked in $G$ (we use well-linked if $\alpha = 1$) if for any two $A, B \subseteq X$ with $|A| = |B|$, the sets $A, B$ are $\alpha$-linked.*

If $X$ is node-well-linked in a graph then $X$ cannot have a sparse node separators. More precisely, if $S$ separates $G - S$ into components and $S$ does not have any vertices of $X$ then no component of $G - S$ can have more than $|S|$ vertices of $X$.

Node-well-linkedness is connected to vertex-expanders. Sometimes people do not distinguish between these two notions too much in the expansion literature because of the following fact: if $G$ is an $\alpha$-edge-expander with maximum degree $d$ then $G$ is also $\Omega(\alpha/d)$-vertex-expander. Thus, if one is working with constant degree graphs the two notions are not very far. However, consider the star graph. One can see that it is an edge-expander but it is very far from being a vertex expander. In fact the largest node-well-linked set in a star is of size 2. In contrast, if you consider the $\sqrt{n} \times \sqrt{n}$ grid then it has a node-well-linked set of size $\Omega(\sqrt{n})$. You should verify this. In fact, no planar graph on $n$ vertices can have a node-well-linked set of size more than $c\sqrt{n}$ for some fixed constant $c$.

Treewidth is an important graph parameter. See `https://en.wikipedia.org/wiki/Treewidth` and pointers. What is the connection between treewidth and well-linked sets?

**Fact 1.** *Let $k$ be the cardinality of the largest node-well-linked set in a graph $G$. Then $k \leq$ treewidth$(G) \leq 4k$.*

Thus, well-linked sets are closely connected to treewidth and in fact most algorithmic approaches to computing treewidth are based on algorithms for sparse node separator computations.

## 2 Expander Decomposition and Well-linked Decomposition

In this section we will prove a simple but important theorem/lemma about decomposing a graph into smaller graphs such that each small graph has good expansion/conductance. The reason for this is that in several settings, the poperties of expanders allow us to obtain good algorithms. If the original graph is an expander we don't have to do anything but what if it is not. In expander decomposition we wish to remove as few edges as possible such that the graph decomposes into expanders. The question is the tradeoff between the number of edges that we remove and the expansion that we can guarantee for the pieces. Technically the process works with conductance but we use the terminology of expander decomposition for historical reasons.

Since one is often interested in finding fast algorithms for expander decomposition it is common to explore tradeoffs between the quality of the conductance of the pieces and the number of edges.

**Definition 6.** *A $(\phi, \epsilon)$-expander decomposition of a (connected) graph $G = (V, E)$ is a decomposition of $G$ into vertex induced subgraphs $G_1 = G[V_1], G_2 = G[V_2], \ldots, G[V_h]$ such that (i) the number of intercluster edges, that is, $\frac{1}{2}\sum_i |\delta(V_i)|$ is at most $\epsilon m$ and (ii) the conductance $\phi_i$ of each $G_i$ is at least $\phi$.*

**Theorem 9.** *Let $G = (V, E)$ be a graph and $c \in (0, 1)$ be a parameter. Suppose there is an $\alpha$-approximation for the* Uniform Sparsest Cut *problem. Then there is an efficient algorithm that outputs an $(\Omega(\frac{c}{\alpha \log m}), c)$-expander decomposition of $G$.*

**Remark 10.** *If we do not care about efficiency we can set $\alpha = 1$. In particular the theorem guarantees that the decomposed pieces have conductance $\Omega(1/\log m)$ while cutting only a constant fraction of the edges. This tradeoff is tight as shown by the hypercube [AALG17].*

We will prove the theorem using a simple algorithm. Since we use a recursive algorithm and the size of the graph changes, we will use $M$ to denote the number of edges in the initial graph which does not change through out the algorithm.

**ExpanderDecomposition**$(G, c)$

1. Assume $G$ is connected (otherwise use algorithm on each connected piece).
   If $|E(G)| \leq (10 \log M)/c$ return $G$
2. Use $\alpha$-approximation algorithm for conductance to find sparsest cut $(S, V - S)$. Let $\phi' = |\delta_G(S)|/\min\{\mathrm{vol(S)}, \mathrm{vol(V-S)}\}$ be its sparsity
3. If $\phi' \geq c/(10 \log M)$ output $G$
4. Else recurse on $G[S]$ and $G[V - S]$ with parameter $c$ and return the union of the decompositions.

**Claim 11.** *The conductance of each cluster output by the algorithm is at least* $\frac{c}{10\alpha \log M}$.

*Proof.* For the base case, if $G$ is connected and has at most $(10 \log M)/c$ edges then conductance of $G$ is at least $c/(10 \log M)$ since at least one edge crosses any cut, and the volume of the smaller side is at most $10 \log M$.

The other case to verify is when the algorithm return $G$ when the sparsity computed by the approximation algorithm is good enough. If the $\alpha$-approximation algorithm outputs a cut $(S, V-S)$ with sparsity at least $c/(10 \log M)$. Since we use an approximation algorithm, the actual sparsity of $G$ cannot be less than $c/(10\alpha \log M)$. ■

We now analyze the total number of edges cut. We do this via a simple recurrence. Let $T(m)$ be the total number of edges cut by the algorithm on a graph with $m$ edges. If $G$ is a constant sized connected graph then the algorithm does not cut any edges. Similarly if $G$ does not have a sparse cut then it does not cut edges. The algorithm removes edges between $S$ and $V - S$ when $|\delta(S)| < \frac{c}{10 \log M}\mathrm{vol(S)}$ where $\mathrm{vol(S)} \leq \mathrm{vol(V-S)}$. Let $m_1$ be the number of edges inside $S$ and let $m_2$ be the number of edges inside $V - S$ and let $|\delta(S)| = m'$. We have $m_1 + m_2 \leq m$ and $m' \leq c(2m_1 + m')/(10 \log M)$. From this it follows that $m' \leq (1 - o(1))\frac{c}{(5 \log M)} m_1 \leq \frac{c}{4 \log m} m_1$. We have $m_1 \leq m_2$ since $S$ is the smaller side. Thus we obtain a recurrence

$$T(m) \leq T(m_1) + T(m_2) + \frac{c}{4 \log M} \min(m_1, m_2)$$

where $m_1 + m_2 \leq m$. One can verify via induction that that $T(m) \leq cm$. Here we use the fact that $M$ is an upper bound on the number of edges through out the recursion.

We can rephrase the theorem in a different form where we want a lower bound on the conductance of the pieces and express the number of edges cut as a function of that parameter.

**Corollary 12.** *Let $G = (V, E)$ be a graph and $\phi$ be a parameter. Suppose there is an $\alpha$-approximation for the* Uniform Sparsest Cut *problem. Then there is an efficient algorithm that computes a $(\phi, O(\alpha \cdot \phi \cdot \log m))$-expander decomposition.*

Note that the number of edges cut is $< m$ only if $\alpha \cdot \log m \cdot \phi < 1$ so one should think of the target $\phi$ as less than $1/(\alpha \log m)$.

**Remark 13.** *The results are phrased in terms of $m$ the number of edges. Capacitated graphs can be handled by scaling since we do not assume that $G$ is simple. However the dependence on $\log m$ means that when capacities are large we are not guaranteed a strongly polynomial bound. However, one can handle this issue in various ways depending on the application. In most applications of expander decomposition it is the case that the total capacity of the edges can be assume to be polynomially bounded in $n$ and in this case the $\log m$ factor is typically replaced with $\log n$.*

**Fast algorithms for expander decomposition:** The computation of expander decompositions is based on sparsest cut algorithms. Traditionally these algoritms were quite slow. There have been several developments in the last few years which enabled sparsest cut to be reduced to a poly-logarithmic number of *s*-*t* flows via the so-called cut-matching game [KRV09, OSVV08] which in turn enabled faster flow algorithms. There are now near-linear time randomized algorithms for expander decomposition (with slightly weaker parameters than the ideal one) for the regimes of interest [SW19]. In some applications the randomized algorithm is not adequate and there has been considerable effort to obtain deterministic algorithms. Here there are almost-linear time algorithms — see [CGL+20, LS21] and pointers. We will see some of these techniques such as the cut-matching game.

## 2.1 Well-linked decomposition

As we saw, well-linked sets allow us to generalize the notion of expansion/conductance from the whole vertex set to subsets of vertices. Similarly one can generalize expander decompositoin to what is called a well-linked decomposition. This material can be skipped for those who are mainly interested in expander decomposition.

We need some set up. Let $G = (V, E)$ be a graph. Suppose we have a multicommodity flow between vertices in $V$ that is routed in $G$. Let $D(uv)$ be the flow routed for the unordered pair $uv$. For each $u \in V$ let $\pi(u) = \sum_{v \in V, v \neq u} D(uv)$ be the total flow from $u$ to some vertex in $X$. We would like the terminals/vertices to be $\pi$-well-linked, but if they are not we would like to do a decomposition into vertex induced subgraphs such that they are. And we would like to do it in such a way that we lose as little amount of the original flow as possible. It is difficult to bound the total flow lost so we will upper bound it by the total number of edges crossing the partitions.

How is this set up related to expander decomposition that we saw earlier? Consider each edge $uv \in E$ and send one unit of flow along that edge for the pair $uv$ then we see that $\pi(u) = \deg(u)$ for each $u \in V$.

**Theorem 14.** *Let $G = (V, E)$ be a graph and $c \in (0, 1)$ be a parameter. Suppose there is an $\alpha$-approximation for the* Uniform Sparsest Cut *problem. Let $\pi : V \to \mathbb{R}_+$ be a weight vector induced by a multicommodity flow in $G$ where $\pi(u)$ is the total flow originating at $u$. Then there is an efficient algorithm that decomposes $G$ into vertex induced subgraphs $G_1 = G[V_1], G_2 = G[V_2], \ldots, G[V_h]$ such that (i) the number of intercluster edges, that is, $\frac{1}{2}\sum_i |\delta(V_i)| \leq c\pi(V)$ and (ii) in each $G_i$, $V_i$ is $\beta\pi'$-well-linked where $\beta = \Omega(\frac{c}{\alpha \log m})$ and $\pi'(u)$ is the total flow incident to $u$ in $G_i$ (that remains from the original flow).*

The algorithm and proof is essentially the same as the one for expander decomposition. Let $M = \pi(V)$ to keep it fixed through out the algorithm as before. We use the $\alpha$-approximation algorithm on $V$ with weights $\pi$. It outputs a cut $(S, V - S)$ with $S$ the smaller side (in terms of $\pi$ weight). If $|\delta(S)| \geq c\pi(S)/(10 \log M)$ we output $G$ and we are guaranteed that $V$ is $\pi/(10\alpha \log M)$-well-linked in $G$ since we used an $\alpha$-approximation algorithm. Otherwise we remove the edges in $\delta(S)$ and recurse on $G[S]$ and $G[V - S]$ after throwing out all the flow that is lost by the edges in $\delta(S)$. The recursion analysis is the same as before we can guarantee that the total number of edges cut is at most $c\pi(V)$.

**Remark 15.** *In routing problems it is more useful to work with the notion of flow-well-linkedness rather than cut-well-linkedness. Theorems similar to the above can be shown via the known flow-cut gap results. We refer the reader to [CKS05].*

**Remark 16.** *The notion of well-linked decomposition can be generalized to node-capacitated setting naturally. It can also be generalized to directed graphs but one has to work with symmetric demands [CE15]. Directed graph expander decompositions have found several applications recently in fast algorithms — see [BGS20] and subsequent work. Via duality they are related to low diameter decompositions in directed graphs which have played a key role in recent fast algorithms for shortest paths with negative lengths.*

**Bibliographic Remarks:** The idea behind expander decomposition is quite simple and was perhaps implicitly used in various ways but people credit the formal definition to work of Goldreich and Ron [GR98] in the context of property testing and to Kannan, Vempala and Vetta [KVV04] in the context of graph clustering. The notion of well-linked sets comes from the literature on treewidth and graph minor theory due to Roberston and Seymour — see Diestel's book on graph theory [Die24]. These notions were adapted by Chekuri, Khanna and Shepherd in their work on routing problems [CKS05] and they developed the notion of well-linked decompositions which have been useful not only in algorithms but also led to results in graph theory [CC16].

# References


[AALG17] Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph clustering using effective resistance. *arXiv preprint arXiv:1711.06530*, 2017.

[BGS20] Aaron Bernstein, Maximilian Probst Gutenberg, and Thatchaphol Saranurak. Deterministic decremental reachability, scc, and shortest paths via directed expanders and congestion balancing. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1123–1134. IEEE, 2020.

[CC16] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *Journal of the ACM (JACM)*, 63(5):1–65, 2016.

[CE15] Chandra Chekuri and Alina Ene. The all-or-nothing flow problem in directed graphs with symmetric demand pairs. *Mathematical Programming*, 154:249–272, 2015.

[CGL+20] Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1158–1167. IEEE, 2020.

[CKS05] Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 183–192, 2005.

[Die24] Reinhard Diestel. *Graph theory.* Springer (print edition); Reinhard Diestel (eBooks), 2024.

[GR98] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. In *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing*, pages 289–298, 1998.

[KRV09] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM (JACM)*, 56(4):1–15, 2009.

[KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.

[LS21] Jason Li and Thatchaphol Saranurak. Deterministic weighted expander decomposition in almost-linear time. *CoRR*, abs/2106.01567, 2021.

[OSVV08] Lorenzo Orecchia, Leonard J Schulman, Umesh V Vazirani, and Nisheeth K Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 461–470, 2008.

[SW19] Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2616–2635. SIAM, 2019.