1 Why Graph Compression?

When we deal with graphs, often times we're interested determining some properties of the graph, e.g., the *diameter*. For an undirected unweighted graph $G = (V, U)$ the diameter D_G is defined as

$$D_G := \max_{u,v \in V} d_G(u, v)$$

Here $d_G(u, v)$ denotes the length of the shortest path between u and v according to the graph G. For example, if G a path on n vertices, the diameter is $n - 1$: (this is the distance from the 'first' and 'last' vertex in this path. For a complete graph K_n on n vertices, the diameter is 1, since the shortest path between any two vertices is just the edge that connects them. For bigger graphs that aren't as nice as paths or cliques, calculating the diameter can involve computing the shortest path between all pairs of vertices. Luckily, we know all-pairs shortest path (APSP) algorithms that take around $O(n \cdot |E|) \leq O(n^3)$ time (remember Johnson's algorithm), which is not too bad.

But what if we don't have that much time? E.g., if we're dealing with a really edge-y graph (like a clique, but we don't know its a clique - because checking if its a clique takes time too). Suppose we are happy with an approximation to the diameter? Maybe then I shouldn't need to look at *all* the edges in the graph to get some idea of what the diameter of the graph is? Maybe I can find a way to "compress" my graph G into a smaller graph H that is a good approximation of G w.r.t the shortest paths.

Our new subgraph H will have vertex set V (we don't want to kick out any vertices - because if v is not in H, there's no way of telling what the shortest path to v is in G) and a much smaller edge set E' (which is some subset of E). This graph will much sparser than G itself. And if we want $d_G(u, v)$, we'll respond with $d_H(u, v)$, which should take much less time to compute if $|E'| \ll |E|$

Doing this cannot guarantee that $d_H(u, v) = d_G(u, v)$. E.g., if G is a clique, removing a single edge that connects u to v to create H will cause $d_H(u, v)$ to jump from 1 to 2. This motivates the following question: Can we find sparse subgraphs H that approximate $d_G(u, v)$ for all u, v? And how fast can we do this?

2 Spanners

A **spanner** of an undirected unweighted graph $G = (V, E)$ is a subgraph $H = (V, E')$ that spans the graph (i.e., vertices connected in G are also connected in H—otherwise calculating shortest paths would be impossible).

Definition 1 *A (k, b)-**spanner** of G is a subgraph H such that for all pairs of vertices u, v:*

$$d_G(u, v) \leq d_H(u, v) \leq k \cdot d_G(u, v) + b$$

*When $b = 0$, this is called a **multiplicative spanner**. When $k = 1$, this is an **additive spanner**.*

Note: the first inequality always holds true: removing edges can never shrink the shortest path. We're really interested in ensuring the upper bound on $d_H(u, v)$, since we don't want to remove so many edges that the graph H is barely recognizable as being derived from G.

Considering a spanner of a graph and settling for an approximation allows us to reduce the time/space used in graph algorithms. (We've seen this sort of trade-off between time/space and quality of the answer before, in say streaming algorithms, or in approximation algorithms.) For instance, if we want the single-source shortest paths in the *unweighted* graph G but settle for an approximation, we can use time and space $O(|E'|+n)$ over $O(|E|+n)$ by running BFS on H rather than G. So if we reduce the number of edges from n^2 to say, $n^{1.5}$, this could be very helpful.

To approximate the diameter of a graph, the spanner bounds will still hold (please prove this for yourself): $D_G \leq D_H \leq k \cdot D_G + b$. We can compute this approximation using time $O(n|E'|)$ as opposed to $O(n|E|)$.

Caveat: we should be mindful of the time it takes to compute the spanner H itself. I.e., if we try to be really careful to ruin as few shortest paths as possible, or use the fewest edges, we may take a lot of time to construct H. So our algorithm should be fast and simple. While we will not discuss this further in this lecture, it is something to keep track of, when thinking about spanners.

3 Multiplicative Spanners

We say a spanner H of G is **k-mutliplicative** if for all pairs of vertices u, v

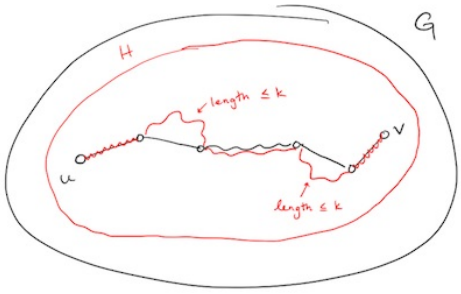
$$d_G(u,v) \leq d_H(u,v) \leq k \cdot d_G(u,v)$$

If we want this to hold true, how many edges do I really need to keep? First, let's make a simple observation that allows us to focus on just edges of G.

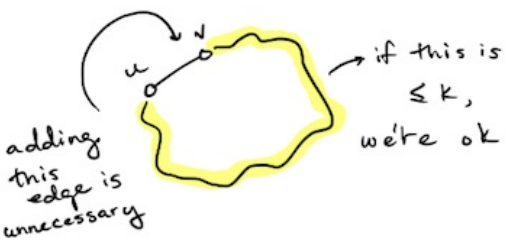
Claim 2 *If for every* edge (u,v) *in G, if the distance between u and v in H is at most k (i.e., $d_H(u,v) \leq k$), then for every pair of vertices $u, v \in G$,*

$$d_H(u,v) \leq k \cdot d_G(u,v).$$

Proof: Fix a shortest path $P = \{e_1, e_2, ..., e_\ell\}$ from u to v in G. It has length $\ell = d_G(u,v)$. Now for each edge $e_i = (v_i, v_{i+1})$ on P there is a path from v_i to v_{i+1} in H of length k. Concatenating these paths in H together gives a path from u to v of length $k \cdot d_G(u,v)$. Now the shortest u-v path in H can be no longer. This proves the claim. ■



What does this observation tell us? As long as I have a path of at most k edges between u and v for every **edge** (u,v) in G, I am a k-multiplicative spanner.



3.1 The Algorithm

If we don't want to add any more edges than necessary, we shouldn't add an edge if we already have a path of length at most k between its endpoints. Another way to phrase this is to say that there will never be a cycle of length $\leq k+1$—if there were, the last added edge on this cycle, the one that created the cycle, would have had a path between its endpoints of length $\leq k$. Creating bigger cycles (of length at least $k+2$) is fine.

This motivates the following greedy algorithm to construct a $k-$multiplicative spanner:

1. Initialize $H = (V, E')$ with $E' = \emptyset$.
2. For every edge $e \in E$, if e doesn't form a cycle of length $\leq k+1$ with other edges in E', add e to E'.

   We can replace this constraint with the equivalent constraint: "For every edge $e \in E$ that connects i to j, if $d_H(i,j) > k$, add e to E', else drop it".

Theorem 3 *H is a k-multiplicative spanner.*

Proof: By Claim 2 we need to just ensure that for every edge of G, there exists a path in H of length at most k. Consider some edge $e = (u, v)$ in G. Either it belongs to H, in which case $d_H(u, v) = 1$. Else it was dropped. By the algorithm description this is because H already contains a path from u to v of length at most k. ■

3.2 How Many Edges?

Ok, so H is a multiplicative spanner. Now we want to figure out how to bound the number of edges we added to E'. First, define the **girth** of a graph G to be the length of the shortest cycle in G.

By the definition of the algorithm, the spanner H is guaranteed to have girth at least $k+2$. Indeed, if there were a cycle of length $k+1$ or less, the last edge on this cycle would never have been added. So, how many edges can I add to a graph so that the girth is no smaller than $k+2$?

To get a sense of this question, consider the question for small values of k:

- How many edges can a graph of girth $k+2 = 3$ have? Girth 3 excludes only parallel edges: a complete graph K_n has girth 3 and $\binom{n}{2}$ edges. So $\Omega(n^2)$.
- For $k = 2$ (girth $k+2 = 4$), we want to exclude 3-cycles. An easy way to exclude 3-cycles is by taking any bipartite graph. A complete bipartite graph (i.e., the bipartite graph with the most edges and many 4-cycles) has $\Omega(n^2)$ edges, and girth 4. So we cannot get a bound better than $\Omega(n^2)$.
- It turns out that if $k = 3$ (girth $k+1 = 5$), we can have at most $O(n^{3/2})$ edges!

In fact we will prove the following theorem soon:

Theorem 4 *For $k = 2t$ or $k = 2t-1$, any graph H with girth at least $k+2$ has $O(n^{1+\frac{1}{t}})$ edges*

Combining this theorem with the observation about the girth of H, we get that a k-multiplicative spanner has $O(n^{1+2/k})$ edges if t is even, and $O(n^{1+2/(k+1)})$ edges if t is odd.

3.3 Proof of Theorem 4

Ok, now to prove that any graph H with large girth has few edges. Here's the general idea. Suppose H has m edges and n vertices.

> We show that H must contain a subgraph H' with almost as many edges as H, but whose minimum degree is high. From this we deduce that n must be large compared to m.

Now the details.

Lemma 5 *Let $\bar{d} := \frac{2m}{n}$ be the* average *degree in a graph H. Then we can construct a nonempty subgraph H' of H with* minimum *degree $\bar{d}/2$.*

Proof: We'll build H' from $H = (V_H, E_H)$ as follows: Let us start with $V_0 = V_H$ and $E_0 = E_H$. We now create a sequence of graphs, which will end with the final graph H'.

1. Let $i = 0$.
2. While there exists a vertex $v \in V_i$ with degree at most $\frac{|E_i|}{|V_i|}$, delete the vertex and all adjacent edges:
   a. $i \leftarrow i + 1$
   b. $V_i \leftarrow V_{i-1} \setminus \{v\}$
   c. $E_i \leftarrow E_{i-1} \setminus \{(v, w) \mid v \sim w\}$.
3. If no such vertex exists, we're done — output $H' = (V_i, E_i)$.

Clearly, H' has the property that it has no vertices of degree $|E_i|/|V_i|$, but how does this quantity compared to $\bar{d}$? *Also, H' has no low-degree vertices, but it could be that H' has no vertices at all!* We will show both things using the same proof. First observe the following fact about real numbers: For $t \leq \frac{x}{y}$, the following holds

$$\frac{x - t}{y - 1} \geq \frac{x - \frac{x}{y}}{y - 1} = \frac{x(1 - \frac{1}{y})}{y(1 - \frac{1}{y})} = \frac{x}{y}$$

Next, observe that the number of edges ℓ we remove is at most $|E_i|/|V_i|$, so the new ratio of edges to vertices is

$$\frac{|E_{i-1}| - \ell}{|V_{i-1}| - 1} = \frac{|E_i|}{|V_i|} \geq \frac{|E_{i-1}|}{|V_{i-1}|}.$$

This chain of logic may be applied repeatedly to get us that

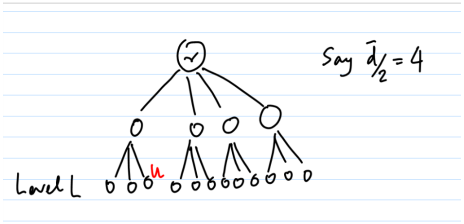
$$\frac{|E_i|}{|V_i|} \geq \frac{|E_{i-1}|}{|V_{i-1}|} \geq \ldots \geq \frac{|E_0|}{|V_0|} = \frac{m}{n} = \frac{\bar{d}}{2} > 0.$$

Since E_i is strictly larger than zero, H' is not empty. Moreover, the stopping condition says the average degree of vertices in H' is at least $|E_i|/|V_i| \geq \bar{d}/2$. ■

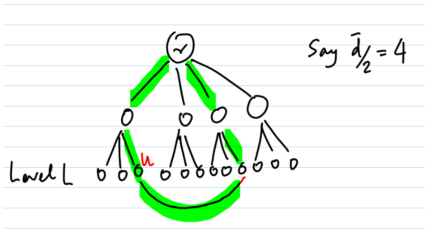
Now, we prove Theorem 4 using this lemma.

Proof: (of Theorem 3) Using Lemma 5, we extract a nonempty subgraph H' of H with minimum degree $\bar{d}/2$. Observe that H' will also have girth at least $k + 2$ since we couldn't have created a smaller cycle by removing vertices or edges.

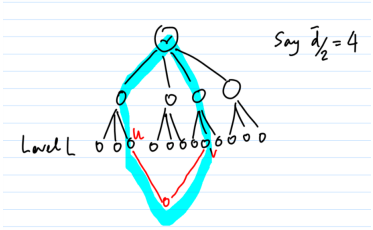
Now fix any vertex as a root, and construct a BFS tree. Consider the nodes at level $L < t$ of the BFS in H'.



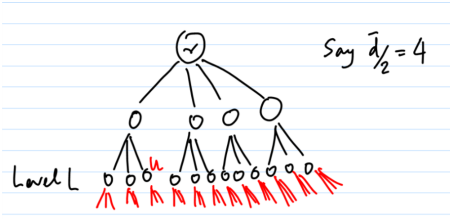
Each such node u has one neighbor at level $L-1$, its parent. Can u have another one of its incident edges to another node already at level L or earlier? No, because then there would be a cycle of length at most $2L - 1 < 2t = k$.



In fact, it cannot even have a neighbor at level $L+1$ in common with some other v, else you would get a cycle of length $2L$.



So all the $\bar{d}/2 - 1$ other neighbors of u must all be disjoint at level $L + 1$. And hence level $L + 1$ must look like this:



So the BFS cannot end earlier, and in fact, the number of nodes at level $L + 1$ are greater than the number at level L by $(\bar{d}/2 - 1)$. This is why the number of nodes at level t in H' is at least $(\bar{d}/2 - 1)^t$.

Since H' has at most n vertices, we get that $n \geq (\frac{\bar{d}}{2} - 1)^t$, and solve for the desired result:

$$\begin{aligned}
\left(\frac{\bar{d}}{2} - 1\right)^t \leq n \quad &\implies \quad \left(\frac{2m/n}{2} - 1\right)^t \leq n \\
&\implies \quad \left(\frac{m}{n} - 1\right)^t \leq n \\
&\implies \quad m \leq n + n^{1+\frac{1}{t}} \quad = O(n^{1+\frac{1}{t}})
\end{aligned}$$

Hence H has at most $O(n^{1+1/t})$ edges, as claimed. ■

3.4 Tight Results?

Now we've proved an upper bound, can we find a matching lower bound? I.e., is it the case that there are graphs with girth $k+2$ where $k = 2t$ or $2t-1$ with $\Omega(n^{1+\frac{1}{t}})$ edges. We don't know: this is a famous open problem called the *Erdös Girth Conjecture*!

If this were true, this means any k-multiplicative spanner H must have $\Omega(n^{1+\frac{1}{t}})$ edges. Why? If we were to delete some edge $\{u, v\}$ from a girth $k+2$ graph G, the distance from u to v would be $k+1$ (since the smallest cycle is $k+2$), so the only k-spanner of a graph G with girth $k+2$ would be G itself.

The girth conjecture on the number of edges is not known for all values of k - so far, its only been proven for $k = 1, 2, 3$ and 5. (Extra credit if you can figure out how to prove it for $k = 4$)!

4 Additive Spanners

Note that the proof above implies that we find a 3-multiplicative spanner with $O(n^{3/2})$ edges. In this section we now give an algorithm to find a 2-additive spanner with almost the same number of edges. Since a 2-additive spanner means

$$d_H(u, v) \leq d_G(u, v) + 2 \leq d_G(u, v) + 2d_G(u, v) = 3d_G(u, v),$$

this is also a 3-multiplicative spanner (and much stronger)!

Our algorithm for finding a 2-additive spanner was to:

1. For each edge $e = \{u, v\} \in E$, if the degree of u or the degree of v is at most $\sqrt{n}$, then include e in E'.
2. Uniformly sample a set S of $2\sqrt{n} \ln n$ vertices of V. For each $v \in S$, grow a shortest path tree T_v in the original graph G, and include all edges in T_v in E'.

4.1 Number of Edges

How many edges do we include in E' in step (1)? Well each edge added in this step is incident to at least one vertex of degree at most $\sqrt{n}$. There are trivially at most n, and each has degree at most $\sqrt{n}$, in total there are at most $n \cdot \sqrt{n}$ edges added in step (1).

How many edges did we include in E' in step (2)? Each shortest path tree T_v has at most $n-1$ edges (even fewer if the graph is disconnected), and there are $|S|$ total shortest path trees that we included, so in total we include at most

$$|S| \cdot (n-1) = (2\sqrt{n} \ln n) \cdot (n-1) = O(n^{3/2} \log n)$$

edges into E' in this step.

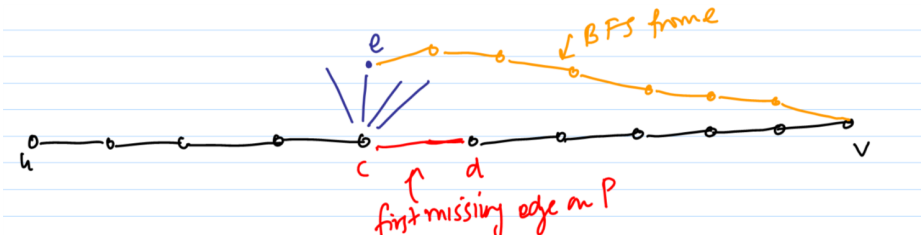
That's the whole algorithm, so in total $|E'|$ has $O(n^{3/2} \log n)$ edges.

4.2 Why is it a 2-additive Spanner?

To prove H is a 2-additive spanner, for any u, v of vertices in V, we want their distance $d_H(u, v)$ is at most $d_G(u, v) + 2$. So let's take a shortest path P from u to v in G (if there is more than one

shortest path, choose an arbitrary one). Then we want to find a u-v path P' in H of length $|P'|$ at most $|P| + 2 = d_G(u, v) + 2$.

If each edge along P is included in step (1) above, the entire path P is in E', and so we can just take $P' = P$, and so there is no loss! Else there is some edge $\{c, d\}$ which is in P but is not included in step (1). Let $\{c, d\}$ be the first edge we see when walking from u to v along P that is not included in step (1). By our algorithm definition, c has degree at least $\sqrt{n}$.



Now we claim that we sample a neighbor e of c in step (2) with very high probability. Why? Each vertex we add to S (suppose we sample vertices with replacement) is not a neighbor of c with probability most

$$1 - \frac{|N(c)|}{n} \leq 1 - \frac{\sqrt{n}}{n} = 1 - \frac{1}{\sqrt{n}},$$

where $N(c) \geq \sqrt{n}$ denotes the neighbors of c. So, the probability that every single one of our vertices in our sample S is not a neighbor of c is

$$\Pr[N(c) \cap S = \emptyset] \leq \left(1 - \frac{1}{\sqrt{n}}\right)^{|S|} \leq e^{-|S|/\sqrt{n}} = e^{-(2\sqrt{n}\ln n)/\sqrt{n}} = e^{-2\ln n} = \frac{1}{n^2}.$$

Consider the "good" event $\mathcal{E}$ that says "for every vertex w of degree at least $\sqrt{n}$, sample S contains a neighbor of w." There are at most n vertices w of degree at least $\sqrt{n}$, and $\Pr[N(w) \cap S = \emptyset] \leq 1/n^2$, by a trivial union bound, $\Pr[\neg\mathcal{E}] \leq n \cdot \frac{1}{n^2} = 1/n$, and therefore $\Pr[\mathcal{E}] \geq 1 - 1/n$.

So conditioned on the good event $\mathcal{E}$ holding, we sampled a vertex e that is a neighbor of c. So in step (2) of our algorithm, we must have included a shortest path tree T_e (with respect to graph G) in our edge set E'. Since one path from node e to v in G is the path which uses edge $\{e, c\}$ followed by the edges along P from c to v, our shortest path tree T_e will contain a e-v path Q of no greater length, i.e., of length $\leq 1 + d_P(c, v)$. (This is the orange path above.) Moreover, $Q \subseteq T_e \subseteq E'$.

Now consider the path P' in H which walks along P from u to vertex c, then takes the edge $\{c, e\}$, and then finally takes the path Q. The length of this path P' is at most $d_P(u, c) + 1 + |Q|$, and we just said $|Q| \leq 1 + d_P(c, v)$. So the length of P' is at most $d_P(u, c) + 1 + 1 + d_P(c, v) = d_G(u, v) + 2$. Since this held for an arbitrary pair of vertices u, v, it holds for all pairs of vertices (conditioned on the event $\mathcal{E}$ above occurring).

Thus, overall, we get a randomized algorithm which takes an arbitrary input graph G, and with probability at least $1 - 1/n$ it outputs a 2-additive spanner with $O(n^{3/2} \log n)$ edges!