1 Graph Spanners

Our goal is to compress information about distances in a graph by looking at distances within a subgraph.

Definition 1.1. An \((\alpha, \beta)\)-spanner of \(G = (V, E)\) is a subgraph \(H = (V, E_H)\), \(E_H \subseteq E\), such that \(\forall u, v \in V\),

\[
d(u, v) \leq d_H(u, v) \leq \alpha d(u, v) + \beta.
\]

If \(\beta = 0\), then it’s a multiplicative spanner (\(\alpha\)-spanner); if \(\alpha = 1\), then it’s an additive spanner (\(+\beta\)-spanner).

In general, directed graphs don’t contain sparse spanners. An example is shown in the figure below, where \(G\) is a directed bipartite graph, with all its edges leaving nodes in set \(U\) and incident on nodes in set \(V\). In this case, any spanner with finite \((\alpha, \beta)\) must contain all edges in \(G\). As a result, we will focus on undirected graphs in this class.

![Diagram of a directed bipartite graph]

2 Multiplicative Spanners

Theorem 2.1. Let \(k \geq 1\) be an integer, then every \(n\)-node graph \(G\) with weighted edges contains a \((2k - 1)\)-spanner with \(O\left(n^{1+\frac{1}{k}}\right)\) edges.

Conjecture 2.1. (Erdős girth conjecture) For integer \(k \geq 1\) and sufficiently large \(n\), there exist \(n\)-node graphs of girth \(\geq 2k + 2\) with \(\Omega\left(n^{1+\frac{1}{k}}\right)\) edges.

Claim 2.1. The Erdős girth conjecture implies that the bound in Theorem 2.1 is tight, and so there exists some graph \(G\) on \(n\) nodes such that any \(2k - 1\) spanner has \(\Omega\left(n^{1+\frac{1}{k}}\right)\) edges.
Proof of Claim 2.1. Let $G$ be an unweighted graph on $n$ edges with girth $2k+2$ and $\Omega\left(n^{1+\frac{1}{k}}\right)$ edges. We’ll show that $G$ has no non-trivial $2k-1$ spanners.

Assume there exists some subgraph $H \neq G$ that is a $2k-1$ spanner for $G$. Choose some edge $(u, v) \in E - E_H$. By the definition of a spanner, $d_H(u, v) \leq (2k-1)d(u, v) = 2k-1$. Therefore there exists some path $P$ in $E_H$ connecting $u, v$ with length at most $2k-1$. However, adding $(u, v)$ to $P$ then completes a cycle in $G$ of length at most $2k$; since $G$ has girth at least $2k+2$, this is a contradiction. This proves the claim.

Proof of Theorem 2.1. We can generate a sufficient $2k-1$ spanner using the Create-Spanner algorithm. We prove the correctness of this algorithm with the following three claims.

Algorithm 1: Create-Spanner($G$)

```plaintext
$E_H \leftarrow \emptyset$.

foreach $(u, v) \in E$ in non-decreasing order do
  if $d_H(u, v) > (2k-1)w(u, v)$ then
    $E_H \leftarrow E_H \cup (u, v)$

Return $H$.
```

Claim 2.2. $H$ is a $(2k-1)$-spanner, i.e., $\forall u, v \in V, d_H(u, v) \leq (2k-1)d(u, v)$.

Claim 2.3. $H$ has girth greater than $2k$.

Claim 2.4. Any $n$-node graph with girth greater than $2k$ has $O\left(n^{1+\frac{1}{k}}\right)$ edges.

Proof of Claim 2.2. Let $u, v$ be vertices in $V$, and $P$ be their shortest path in $G$. For each edge $(x, y)$ in $P$, either:
- $(x, y) \in E_H$
- There is some path in $H$ between $x, y$ of length at most $(2k-1)w(x, y)$. If no such path exists, then $(x, y)$ would have been added to $E_H$ in Create-Spanner when it was considered.

Therefore

$$d_H(u, v) \leq \sum_{(x, y) \in P} d_H(x, y) \leq \sum_{(x, y) \in P} (2k-1)w(x, y) = (2k-1)w(P) = (2k-1)d(u, v).$$

□

Proof of Claim 2.3. Assume $H$ has a cycle $C$ of length $\leq 2k$. Let $(u, v)$ be the highest weighted edge of $C$. Then $(u, v)$ is the last edge in $C$ added to $E_H$, satisfying

$$\sum_{(x, y) \in C, (x, y) \neq (u, v)} w(x, y) > (2k-1)w(u, v)$$

On the other hand, the path $C \setminus (u, v)$ has length at most $(2k-1)w(u, v)$, giving

$$\sum_{(x, y) \in C, (x, y) \neq (u, v)} w(x, y) \leq (2k-1)w(u, v)$$

This gives a contradiction. □
Proof of Claim 2.4. Let $H$ be any graph with girth greater than $2k$ and at least $10n^{1+\frac{1}{k}}$ edges. Modify the graph by repeatedly removing any nodes of degree $\leq \lceil n^{\frac{1}{k}} \rceil$, and any edges incident to that node, until no such nodes exist. The total number of edges removed in this way at most $2n \cdot \lceil n^{\frac{1}{k}} \rceil$, which means that at least $8n^{1+\frac{1}{k}}$ edges remain (and so the graph is not empty).

The minimum degree of the resulting subgraph is greater than $\lceil n^{\frac{1}{k}} \rceil$. However, by homework problem 1.3b, this means that the subgraph has girth at most $2k$, and therefore the original graph does as well. This is a contradiction.

The subgraph returned by Create-Spanner is a $(2k-1)$-spanner by Claim 2.2, and has $O\left(n^{1+\frac{1}{k}}\right)$ edges by Claim 2.3, Claim 2.4. This completes the proof of the theorem.

3 Additive Spanners

We restrict ourselves to unweighted graphs, in which an additive spanner makes sense.

**Theorem 3.1.** Any $n$-node graph $G$ has a $+2$-spanner with $O\left(n^2 \log n\right)$ edges.

**Theorem 3.2.** Any $n$-node graph $G$ has a $+4$-spanner with $\tilde{O}\left(n^{\frac{2}{3}}\right)$ edges.

**Theorem 3.3.** Any $n$-node graph $G$ has a $+6$-spanner with $\tilde{O}\left(n^{\frac{4}{3}}\right)$ edges.

**Theorem 3.4.** (Abboud and Bodwin ’15) There is no additive spanner on $O\left(n^{\frac{2}{3}} - \epsilon\right)$, $\epsilon > 0$ edges for $n$-node graphs.

**Proof of Theorem 3.1.**

The proof of this theorem is very similar to the $+2$-approximation to the APSP problem covered in Lecture 4. Let $S$ be a hitting set for $\{N(v) \mid \deg(v) \geq \sqrt{n}\}$. From Lecture 3, we know that we can choose $S$ with $|S| = O(\sqrt{n} \log n)$. Do a BFS search from each $s \in S$, and add the resulting $n$ edges to $E_H$. For every $u \in V$ with $\deg(u) < \sqrt{n}$, add all edges incident to $u$ to $E_H$. By construction, $E_H = O(|S| \cdot n) + O(n\sqrt{n}) = O\left(n^2 \log n\right)$. Consider any pair of edges $(u, v) \in V$ with shortest path $P$ in $G$. We have two cases:

- $P$ contains only low-degree nodes. Then $P$ is entirely contained in $E_H$, so $d_H(u, v) = d(u, v)$.
- $P$ contains a high-degree node $x$. Let $s \in S$ be a node adjacent to $x$. Then we can approximate the distance from $u$ to $v$ by appending the paths $(u, s), (v, s)$, since $E_H$ contains shortest paths from $s$ to every other element. Thus
  \[d_H(u, v) \leq d_H(u, s) + d_H(v, s) = d(u, s) + d(v, s) \leq (d(u, x) + 1) + (d(v, x) + 1) = d(u, v) + 2.\]

Therefore the $H$ constructed by this algorithm is a $+2$-spanner, as desired.

We can improve on the size of this spanner slightly with worse constant errors.
Proof of Theorem 3.2. Today we will show part of the algorithm and explain the reason behind those steps.

**Algorithm 2: +4-Spanner-Partial(G)**

- **Step 1:** Add all edges incident to low degree nodes (degree ≤ D) to H.
- **Step 2:** Let T be a random subset of nodes of size \( \Theta(\frac{n}{DL} \log n) \). For each \( t \in T \), place the entire BFS(t) to H.
- **Step 3:** Deal with the remaining case.

**Step 1:** Deals with the case where the shortest path consists of only low degree nodes. We add at most \( O(nD) \) edges to H.

**Step 2:** This deals with the case where P is a \((u,v)\)-shortest path with \( \geq L \) high-degree nodes. By Claim 3.1, we know that a hitting set of size \( O(\frac{n}{DL}) \) will contain a node with a neighbor in P with high probability. This being the case, we will have:

\[
d_H(u,v) \leq d_H(u,t) + d_H(t,v) \leq d(u,v) + 2.
\]

This step adds \( O(\frac{n^5}{DL}) \) edges.

**Step 3:** Deals with the case where the number of high-degree nodes is between 1 and L.

**Claim 3.1.** Let \( P \) be a \((u,v)\)-shortest path with \( \geq L \) high-degree nodes, then the neighborhood of \( P \), \( N(P) \) has \( |N(P)| \geq \frac{DL}{3} \).

**Proof of Claim 3.1.** Let node \( x \) be a node not on \( P \). Then \( x \) can have at most 3 neighbors on this path; otherwise \( P \) can be shortened by routing through \( x \). This means that there are at least \( \frac{DL}{3} \) vertices adjacent to \( P \).

Therefore we can choose a hitting set \( S \) of size \( O\left(\frac{n^5}{DL} \log n\right) \) such that any for such pair of edges \((u,v)\) and the corresponding shortest path \( P \), \( P \) is adjacent to some \( s \in S \). We can then perform a BFS for every \( s \in S \) and add all the resulting edges to \( E_H \).

References