Approximating the Norms of Graph Spanners

Eden Chlamtáč
אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Michael Dinitz
Johns Hopkins University

Thomas Robinson
אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

APPROX 2019

Graph Spanners: Basics

This talk is about *spanners*

Given graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, subgraph $\mathbf{H}$ of $\mathbf{G}$ is a $\mathbf{t}$-*spanner* of $\mathbf{G}$ if

$$\mathbf{d_H}(\mathbf{u}, \mathbf{v}) \leq \mathbf{t} \cdot \mathbf{d_G}(\mathbf{u}, \mathbf{v}) \quad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}$$

- $\mathbf{t}$ is the *stretch* of the spanner.
- In this paper: $\mathbf{G}$ undirected, unweighted, connected
- Sufficient for stretch condition to hold for all edges $\{\mathbf{u}, \mathbf{v}\} \in \mathbf{E}$

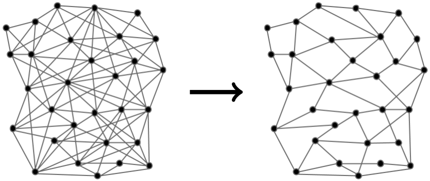
Graph Spanners: Basics

This talk is about *spanners*

Given graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, subgraph $\mathbf{H}$ of $\mathbf{G}$ is a $\mathbf{t}$-*spanner* of $\mathbf{G}$ if

$$\mathbf{d_H}(\mathbf{u}, \mathbf{v}) \leq \mathbf{t} \cdot \mathbf{d_G}(\mathbf{u}, \mathbf{v}) \quad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}$$

- $\mathbf{t}$ is the *stretch* of the spanner.
- In this paper: $\mathbf{G}$ undirected, unweighted, connected
- Sufficient for stretch condition to hold for all edges $\{\mathbf{u}, \mathbf{v}\} \in \mathbf{E}$



Original graph **G** → A **3**-spanner **H**

Classical Objectives

Want to have small stretch, and small “cost”.

Two natural cost measures: total # edges, maximum degree.

Classical Objectives

Want to have small stretch, and small "cost".
Two natural cost measures: total # edges, maximum degree.

 # edges: [Althöfer et al '93]:

- For any positive integer $\mathbf{k}$, all graphs have a $\mathbf{(2k-1)}$-spanner with $\mathbf{O(n^{1+1/k})}$ edges, and

Classical Objectives

Want to have small stretch, and small "cost".
Two natural cost measures: total # edges, maximum degree.

 # edges: [Althöfer et al '93]:

- For any positive integer $\mathbf{k}$, all graphs have a $\mathbf{(2k-1)}$-spanner with $\mathbf{O(n^{1+1/k})}$ edges, and
- There exist graphs in which all $\mathbf{(2k-1)}$-spanners have $\mathbf{\Omega(n^{1+1/k})}$ edges (assuming **Erdös Girth Conjecture**).

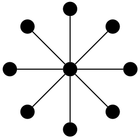
Classical Objectives

Want to have small stretch, and small "cost".
Two natural cost measures: total # edges, maximum degree.

 # edges: [Althöfer et al '93]:

- For any positive integer $\mathbf{k}$, all graphs have a $\mathbf{(2k-1)}$-spanner with $\mathbf{O(n^{1+1/k})}$ edges, and
- There exist graphs in which all $\mathbf{(2k-1)}$-spanners have $\mathbf{\Omega(n^{1+1/k})}$ edges (assuming **Erdös Girth Conjecture**).

No such theorem possible for max degree! Star graph.
Removing any edge cases infinite stretch



Optimizing Spanners

Switch our point of view from tradeoffs to optimization.
Given $\mathbf{G}, \mathbf{k}$, efficient algorithm for finding *best* $\mathbf{t}$-spanner of $\mathbf{G}$?

Optimizing Spanners

Switch our point of view from tradeoffs to optimization.
Given $\mathbf{G}, \mathbf{k}$, efficient algorithm for finding *best* $\mathbf{t}$-spanner of $\mathbf{G}$?

- BASIC $\mathbf{t}$-SPANNER: "best" = fewest edges
  - Lots known – come chat with me!
  - High-level view: can't really beat trivial $\mathbf{O(n^{1/k})}$-approximation for $\mathbf{t = 2k - 1}$.
  - Can slightly in some special cases: $\mathbf{t = 3}$ [BBMRY '13] and $\mathbf{t = 4}$ [D-Zhang '16]
- LOWEST DEGREE $\mathbf{t}$-SPANNER (LD$\mathbf{t}$S): "best" = min max degree
  - Chlamtáč-D '16: $\mathbf{O(\Delta^{(1-\frac{1}{t})^2})}$-approximation, $\mathbf{\Omega(\Delta^{1/t})}$ lower bound
  - Chlamtáč-D-Krauthgamer '12: $\mathbf{\tilde{O}(\Delta^{3-2\sqrt{2}})}$-approx when $\mathbf{t = 2}$ (Sherali-Adams)

Classical Objectives: Motivation and Issues

- Number of Edges:
  - Pros: natural objective, very nice tradeoff theorems. Well-studied. Often what's needed in applications.
  - Cons: Do we really not care if one node has huge degree, as long as others small? Load in distributed settings?
- Maximum Degree:
  - Pros: Encourages low loads in distributed settings. Natural objective.
  - Cons: If some node forced to have large degree, do we really want to allow all other nodes to have large degree?

Classical Objectives: Motivation and Issues

- Number of Edges:
  - Pros: natural objective, very nice tradeoff theorems. Well-studied. Often what's needed in applications.
  - Cons: Do we really not care if one node has huge degree, as long as others small? Load in distributed settings?
- Maximum Degree:
  - Pros: Encourages low loads in distributed settings. Natural objective.
  - Cons: If some node forced to have large degree, do we really want to allow all other nodes to have large degree?

- Want something new: encourages max degree to be small, but also encourages other nodes to have small degree even if max forced to be large.

New Objective

- Observation: consider vector $\mathbf{d_G} \in \mathbb{Z}^{\mathbf{n}}_{\geq \mathbf{0}}$ of vertex degrees in $\mathbf{G}$.
  - Number of edges is $\frac{1}{2}\|\mathbf{d_G}\|_1$
  - Maximum degree is $\|\mathbf{d_G}\|_\infty$
- Interpolate between the two!

New Objective

- Observation: consider vector $\mathbf{d_G} \in \mathbb{Z}^{\mathbf{n}}_{\geq 0}$ of vertex degrees in $\mathbf{G}$.
  - Number of edges is $\frac{1}{2}\|\mathbf{d_G}\|_1$
  - Maximum degree is $\|\mathbf{d_G}\|_\infty$
- Interpolate between the two!

The ℓ_p-norm objective is to minimize

$$\|\mathbf{H}\|_p = \|\mathbf{d_H}\|_p = \left(\sum_{u \in V} \mathbf{d_H}(u)^p\right)^{1/p}$$

- For $1 < p < \infty$, encourages both sparsity and low maximum degree!
  - Standard objective in clustering, scheduling, etc.

ℓ_p-Objective: Tradeoffs

Introduced this objective in [Chlamtáč-D-Robinson ICALP '19]

Theorem: For every $\mathbf{k, p \geq 1}$, every graph admits a $\mathbf{(2k-1)}$-spanner with ℓ_p-norm $\mathbf{\max(O(n), O(n^{\frac{k+p}{kp}}))}$. This bound is also tight.

ℓ_p-Objective: Tradeoffs

Introduced this objective in [Chlamtáč-D-Robinson ICALP '19]

Theorem: For every $k, p \geq 1$, every graph admits a $(2k-1)$-spanner with ℓ_p-norm $\max(O(n), O(n^{\frac{k+p}{kp}}))$. This bound is also tight.

Solved the tradeoff question, but what about optimization?

Definition: In the MINIMUM ℓ_p-NORM t-SPANNER problem, we are given p, t, G, and our goal is to find the t-spanner H of G minimizing $\|H\|_p$

Focus of this paper, with $p = 2, t = 3$ (some results generalizable)

Results

First, study greedy algorithm (used to prove tradeoffs).

Greedy is an $\tilde{O}(n^{3/7})$-approximation for Minimum ℓ_2-Norm 3-Spanner (and this is tight).

New algorithm based on rounding convex relaxation.

There is an $\tilde{O}(n^{5/13})$-approximation for Minimum ℓ_2-Norm 3-Spanner.

Hardness result (more careful analysis of max-degree hardness).

Unless $\mathbf{NP} \subseteq \mathbf{BPTIME}(2^{\mathrm{polylog}(n)})$, for any $\epsilon > 0$ there is no polynomial-time algorithm for Minimum ℓ_2-Norm 3-Spanner with approximation ratio better than $2^{\log^{1-\epsilon} n}$.

Why greedy?

Since we have a better algorithm, why study greedy?

Why greedy?

Since we have a better algorithm, why study greedy?

- Natural and important algorithm – should understand its performance!
- Demonstrates that ℓ_2 norm for stretch $\mathbf{3}$ is *fundamentally different* from ℓ_1 or ℓ_∞ in terms of greedy approximation.

Why greedy?

Since we have a better algorithm, why study greedy?

- Natural and important algorithm – should understand its performance!
- Demonstrates that ℓ_2 norm for stretch $\mathbf{3}$ is *fundamentally different* from ℓ_1 or ℓ_∞ in terms of greedy approximation.
  - ℓ_1: Greedy always has at most $\mathbf{O(n^{3/2})}$ edges, so trivially an $\mathbf{O(n^{1/2})}$-approximation. Tight.
  - ℓ_∞: Greedy has max degree at most $\mathbf{\Delta}$, and $\mathbf{OPT} \geq \mathbf{\Delta^{1/3}}$. So $\mathbf{O(\Delta^{2/3})} = \mathbf{O(n^{2/3})}$-approximation. Tight.

Why greedy?

Since we have a better algorithm, why study greedy?

- Natural and important algorithm – should understand its performance!
- Demonstrates that ℓ_2 norm for stretch $\mathbf{3}$ is *fundamentally different* from ℓ_1 or ℓ_∞ in terms of greedy approximation.
  - ℓ_1: Greedy always has at most $\mathbf{O(n^{3/2})}$ edges, so trivially an $\mathbf{O(n^{1/2})}$-approximation. Tight.
  - ℓ_∞: Greedy has max degree at most $\mathbf{\Delta}$, and $\mathbf{OPT} \geq \mathbf{\Delta^{1/3}}$. So $\mathbf{O(\Delta^{2/3}) = O(n^{2/3})}$-approximation. Tight.
  - ℓ_2: Greedy has at most $\mathbf{O(n)}$ edges (tight), $\mathbf{OPT} \geq \mathbf{\Omega(n^{1/2})}$ (tight). But greedy is $\mathbf{\tilde{O}(n^{3/7})}$-approximation!

Why greedy?

Since we have a better algorithm, why study greedy?

- Natural and important algorithm – should understand its performance!
- Demonstrates that ℓ_2 norm for stretch $\mathbf{3}$ is *fundamentally different* from ℓ_1 or ℓ_∞ in terms of greedy approximation.
  - ℓ_1: Greedy always has at most $\mathbf{O(n^{3/2})}$ edges, so trivially an $\mathbf{O(n^{1/2})}$-approximation. Tight.
  - ℓ_∞: Greedy has max degree at most $\mathbf{\Delta}$, and $\mathbf{OPT} \geq \mathbf{\Delta^{1/3}}$. So $\mathbf{O(\Delta^{2/3})} = \mathbf{O(n^{2/3})}$-approximation. Tight.
  - ℓ_2: Greedy has at most $\mathbf{O(n)}$ edges (tight), $\mathbf{OPT} \geq \mathbf{\Omega(n^{1/2})}$ (tight). But greedy is $\mathbf{\tilde{O}(n^{3/7})}$-approximation!
- Approximation ratio of greedy cannot be determined by "absolute" guarantees for $\mathbf{p = 2}$, unlike $\mathbf{p = 1, \infty}$!

Why greedy?

Since we have a better algorithm, why study greedy?

- Natural and important algorithm – should understand its performance!
- Demonstrates that ℓ_2 norm for stretch 3 is *fundamentally different* from ℓ_1 or ℓ_∞ in terms of greedy approximation.
  - ℓ_1: Greedy always has at most $O(n^{3/2})$ edges, so trivially an $O(n^{1/2})$-approximation. Tight.
  - ℓ_∞: Greedy has max degree at most Δ, and $OPT \geq \Delta^{1/3}$. So $O(\Delta^{2/3}) = O(n^{2/3})$-approximation. Tight.
  - ℓ_2: Greedy has at most $O(n)$ edges (tight), $OPT \geq \Omega(n^{1/2})$ (tight). But greedy is $\tilde{O}(n^{3/7})$-approximation!
- Approximation ratio of greedy cannot be determined by "absolute" guarantees for $p = 2$, unlike $p = 1, \infty$!
- Interesting analysis: write a constant-size LP, argue it characterizes approximation ratio, give tight bound on LP.

Approximation Algorithm: Convex Relaxation

- Let $\mathcal{P}(u, v)$ be all $u \rightsquigarrow v$ paths of length at most 3

$$
\begin{aligned}
\min \quad & \left(\sum_{v \in V} \left(\sum_{e \sim v} x_e \right)^2 \right)^{1/2} & \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}(u,v)} y_p = 1 & \forall (u, v) \in E \\
& x_e \geq \sum_{p \in \mathcal{P}(u,v): e \in p} y_p & \forall (u, v), e \in E \\
& x_e, y_p \geq 0 & \forall e, p
\end{aligned}
$$

- Standard network design LP relaxation, except non-linear objective
    - Easily solved with (e.g.) Ellipsoid
- Use two different rounding algorithms, trade them both off with greedy

Rounding Algorithm 1

Super simple rounding algorithm:

- Add each $\mathbf{e} \in \mathbf{E}$ to $\mathbf{H_1}$ independently with probability $\mathbf{x_e^{3/7}}$

Rounding Algorithm 1

Super simple rounding algorithm:

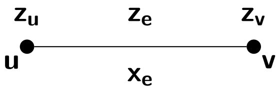
- Add each $\mathbf{e} \in \mathbf{E}$ to $\mathbf{H_1}$ independently with probability $\mathbf{x_e^{3/7}}$

Problem: might not result in a spanner.

If add with probability $\mathbf{x_e^{1/3}}$, would be a spanner, would exactly be algorithm for ℓ_∞ objective from [Chlamtáč-D '16]

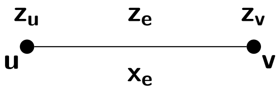
Rounding Algorithm 2 (Simplified)

- For each $\mathbf{u} \in \mathbf{V}$, draw $\mathbf{z_u} \in_R [0,1]$ u.a.r.
- For each $\mathbf{e} \in \mathbf{E}$, draw $\mathbf{z_e} \in_R [0,1]$ u.a.r.
- Add $\mathbf{e} = \{\mathbf{u}, \mathbf{v}\}$ to $\mathbf{H_2}$ if at least one of the following conditions holds:
  - $\mathbf{z_u} \leq \mathbf{x_e}^{1/4}$ and $\mathbf{z_v} \leq \mathbf{x_e}^{1/4}$, or
  - $\mathbf{z_u} \leq \mathbf{x_e}^{1/4}$ and $\mathbf{z_e} \leq \mathbf{x_e}^{1/4}$, or
  - $\mathbf{z_v} \leq \mathbf{x_e}^{1/4}$ and $\mathbf{z_e} \leq \mathbf{x_e}^{1/4}$.



Rounding Algorithm 2 (Simplified)

- For each $\mathbf{u} \in \mathbf{V}$, draw $\mathbf{z_u} \in_R [0,1]$ u.a.r.
- For each $\mathbf{e} \in \mathbf{E}$, draw $\mathbf{z_e} \in_R [0,1]$ u.a.r.
- Add $\mathbf{e} = \{\mathbf{u}, \mathbf{v}\}$ to $\mathbf{H_2}$ if at least one of the following conditions holds:
  - $\mathbf{z_u} \le \mathbf{x_e^{1/4}}$ and $\mathbf{z_v} \le \mathbf{x_e^{1/4}}$, or
  - $\mathbf{z_u} \le \mathbf{x_e^{1/4}}$ and $\mathbf{z_e} \le \mathbf{x_e^{1/4}}$, or
  - $\mathbf{z_v} \le \mathbf{x_e^{1/4}}$ and $\mathbf{z_e} \le \mathbf{x_e^{1/4}}$.

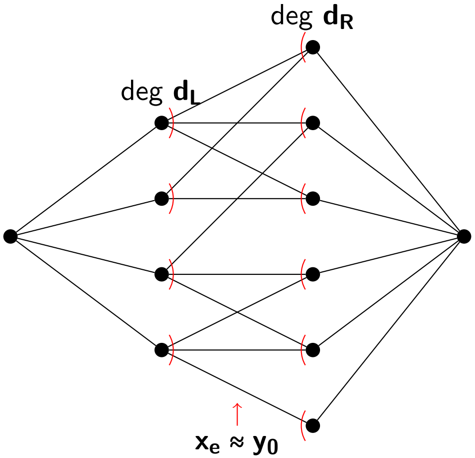


New aspect: rounds based on randomness at both vertices *and* edges

- Sampling at edges: [D-Krauthgamer '11, BBMRY '13, Chlamtáč-D '16]
- Sampling at vertices [D-Krauthgamer '11, D-Zhang '16].
- First algorithm that does both (?)

Use [Chlamtáč-D '16]:

- Bucket and prune $\mathbf{u} \leadsto \mathbf{v}$ paths
- Get that WLOG, LP solution very regular:
- Loses some polylogs



Correctness

Fix $\{\mathbf{u},\mathbf{v}\} \in \mathbf{E}$.

Lemma: If $\mathbf{max(d_L, d_R)} \geq \tilde{\Omega}(\mathbf{y_0}^{-2/3})$, then Rounding Algorithm 1 will include some $\mathbf{p} \in \mathcal{P}(\mathbf{u},\mathbf{v})$ with probability $\tilde{\Omega}(\mathbf{1})$.

Lemma: If $\mathbf{d_L}, \mathbf{d_R} \leq \tilde{O}(\mathbf{y_0}^{-2/3})$, then Rounding Algorithm 2 will include some $\mathbf{p} \in \mathcal{P}(\mathbf{u},\mathbf{v})$ with probability $\tilde{\Omega}(\mathbf{1})$

So repeat $\tilde{O}(\mathbf{1})$ times, get high probability bounds.
Union bound over all $\{\mathbf{u},\mathbf{v}\} \in \mathbf{E}$.

Correctness: Intuition

Modified Algorithm 1: choose each edge $\mathbf{e}$ independently w.p. $\mathbf{x_e^{1/3}}$ (instead of $\mathbf{x_e^{3/7}}$)

- Get path $\mathbf{p} = (\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})$ with probability

$$(\mathbf{x_{e_1} x_{e_2} x_{e_3}})^{1/3} \geq (\min(\mathbf{x_{e_1} x_{e_2} x_{e_3}}))^{1} \geq \mathbf{y_p}$$

- So get each path with the "right" probability, so in expectation get at least one path since $\sum_{\mathbf{p} \in \mathcal{P}(\mathbf{u},\mathbf{v})} \mathbf{y_p} = \mathbf{1}$
  - Issue: Paths not disjoint! Concentration?

Correctness: Intuition

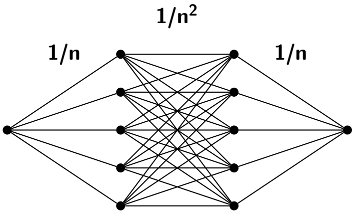
Modified Algorithm 1: choose each edge $\mathbf{e}$ independently w.p. $\mathbf{x_e^{1/3}}$ (instead of $\mathbf{x_e^{3/7}}$)

- Get path $\mathbf{p} = (\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})$ with probability

$$(x_{e_1} x_{e_2} x_{e_3})^{1/3} \geq (\min(x_{e_1} x_{e_2} x_{e_3}))^1 \geq y_p$$

- So get each path with the "right" probability, so in expectation get at least one path since $\sum_{p \in \mathcal{P}(u,v)} y_p = 1$
  - Issue: Paths not disjoint! Concentration?

- Intuition of [Chlamtáč-D '16] : if paths not disjoint, actually doing much better!
  - Get $\mathbf{n(1/n)^{1/3} = n^{2/3}}$ left edges, $\mathbf{n^{2/3}}$ right edges
  - $\mathbf{n^{4/3}}$ ways to complete a path, get each w.p. $\mathbf{1/n^{2/3}}$
  - So get about $\mathbf{n^{2/3}}$ paths!



Correctness: Intuition

Decrease sampling probability to $\mathbf{x_e^{3/7}}$.

- If paths overlap a lot ($\mathbf{\max(d_L, d_R) \geq \tilde{\Omega}(y_0^{-2/3})}$), Rounding Alg 1 still works.
- If not, do something else: correlate at nodes!
- Can't do this for ℓ_∞-metric, but (in this case) can do this for ℓ_2-metric.

Correctness: Intuition

Decrease sampling probability to $\mathbf{x_e^{3/7}}$.

- If paths overlap a lot ($\mathbf{max(d_L, d_R) \geq \tilde{\Omega}(y_0^{-2/3})}$), Rounding Alg 1 still works.
- If not, do something else: correlate at nodes!
- Can't do this for ℓ_∞-metric, but (in this case) can do this for ℓ_2-metric.
  - Having edges bought *only* by nodes has too much correlation, ends up with large degrees.
  - Need to mix edges paying for themselves (randomness at edges) with being bought by endpoints (randomness at nodes)
  - Argue that if paths "mostly disjoint", works well in expectation, and can prove concentration.

Conclusion & Open Questions

- For stretch **3**, ℓ_2-norm: analyzed greedy (tight), hardness of approximation, complicated algorithm to beat greedy.

Conclusion & Open Questions

- For stretch $\mathbf{3}$, ℓ_2-norm: analyzed greedy (tight), hardness of approximation, complicated algorithm to beat greedy.
- What about other $\mathbf{p}$, other stretch?
  - Some things generalize.
    - Hardness
    - Analysis of greedy should (some really annoying technicalities)
    - Algorithm 2 should generalize to other $\mathbf{p}$
  - Some don't
    - Better than greedy for stretch $> \mathbf{3}$?
- Even for $\mathbf{p} = \mathbf{2}, \mathbf{k} = \mathbf{3}$, gap between upper bound and hardness. Better algorithms?
- What about $\ell_{\mathbf{p}}$-norm of degree vector for other network design problems?

Conclusion & Open Questions

- For stretch $\mathbf{3}$, ℓ_2-norm: analyzed greedy (tight), hardness of approximation, complicated algorithm to beat greedy.
- What about other $\mathbf{p}$, other stretch?
  - Some things generalize.
    - Hardness
    - Analysis of greedy should (some really annoying technicalities)
    - Algorithm 2 should generalize to other $\mathbf{p}$
  - Some don't
    - Better than greedy for stretch $> \mathbf{3}$?
- Even for $\mathbf{p} = \mathbf{2}, \mathbf{k} = \mathbf{3}$, gap between upper bound and hardness. Better algorithms?
- What about $\ell_{\mathbf{p}}$-norm of degree vector for other network design problems?

Thanks!