Multi-Level Graph Spanners (CSC 620)

Abu Reyan Ahmed, Indrayudh Roy

May 4, 2018

Overview

Introduction

- Given a graph $G(V, E)$, a subgraph $G'(V, E')$ is a t-spanner of G, if for every $u, v \in V$, the distance from u to v in G' is at most t times longer than the distance in G. We refer to t as the *stretch factor* of G'.

Introduction

- Given a graph $G(V, E)$, a subgraph $G'(V, E')$ is a t-spanner of G, if for every $u, v \in V$, the distance from u to v in G' is at most t times longer than the distance in G. We refer to t as the *stretch factor* of G'.
- Peleg et al. [1] have shown that this is an NP-complete problem.

Introduction

- Given a graph $G(V, E)$, a subgraph $G'(V, E')$ is a t-spanner of G, if for every $u, v \in V$, the distance from u to v in G' is at most t times longer than the distance in G. We refer to t as the *stretch factor* of G'.
- Peleg et al. [1] have shown that this is an NP-complete problem.
- A reduction from the Edge Dominating Set problem for bipartite graph.

MLST Problem

Definition 1 (Multi-Level Spanner Problem)

Given a sequence of positive integers $t_1, t_2, \ldots, t_l$ (Note that we have not mentioned the relationship among the stretch factors. We will consider either $t_1 < t_2 < \ldots < t_l$ or $t_1 > t_2 > \ldots > t_l$.), a multi-level $(t_1, t_2, \ldots, t_l)$-spanner of G is a set of spanners $\{t_i\text{-spanner} : i \in \{1, 2, \ldots, l\}\}$ such that t_i-spanner $\subseteq t_{i+1}$-spanner and $\sum_{i \in \{1,2,\ldots,l\}} c(t_i\text{-spanner})$ is minimum where $c(G')$ is the sum of the weights of all edges of G' for weighted graphs and for unweighted graphs it is the number of edges.

Multi-Level Spanners

Definition 2 (Multi-Level Spanner with Nested Terminal Sets)

Given an input graph G, a sequence of positive integers $t_1, t_2, \ldots, t_l$ ($t_1 < t_2 < \ldots < t_l$ or $t_1 > t_2 > \ldots > t_l$) and l nested terminal sets $T_1 \subset T_2 \subset \ldots \subset T_l \subset V$, a $(t_1, t_2, \ldots, t_l)$-spanner with nested terminal sets is a series of spanners $G_1, G_2, \ldots, G_l$ such that

1. The stretch factor of G_i is equal to t_i
2. $G_{i-1} \subseteq G_i$
3. The vertex set of G_i contains T_i
4. The sum $\sum_{i \in \{1,2,\ldots,l\}} c(t_i\text{-spanner})$ is minimum.

Multi-Level Spanners

Definition 3 (Additive Spanner)

Let $G(V, E)$ be an unweighted graph. We denote the shortest path distance between two vertices $u, v \in V$ in G by $d_G(u, v)$. A subgraph $G'(V, E')$ is an additive t-spanner of G, if for every $u, v \in V$, $d_{G'}(u, v) \leq d_G(u, v) + t$. We refer to t as the *additive stretch factor* of G'.

Multi-Level Spanners

Definition 4 (Multi-Level Additive Spanner)

Given an input graph G, an integer $t_1 < t_2 < \cdots < t_l$ and l nested terminal sets $T_1 \subset T_2 \subset \ldots \subset T_l \subset V$, a multi-level additive $(t_1, t_2, \cdots, t_l)$-spanner is a series of spanners $G_1, G_2, \ldots, G_l$ such that

1. The additive stretch factor of G_i is equal to t_i
2. $G_{i-1} \subseteq G_i$
3. The vertex set of G_i contains T_i
4. The sum $\sum_{i \in \{1,2,\ldots,l\}} c(t_i\text{-spanner})$ is minimum.

Multi-Level Spanners

Definition 5 (A Simple Multi-level Additive Spanner)

1. Every level have same additive stretch: $t_1 = t_2 = \cdots = t_l = 2$.
2. Also there is a pattern on the size of the terminal sets. The size of the terminal sets increase exponentially as we go from top to bottom: $|T_1| = 2, |T_2| = 4, \ldots, |T_l| = 2^l$.

Known Results

- Cygan et al. [2] have shown how to construct $O(n\sqrt{|S|})$ size subset-wise additive spanner for stretch 2 where $|S|$ is the size of the subset.
- Kavitha [3] has studied pairwise spanners with additive stretch factors 4 and 6.

Compute Independent Spanners

- First we assume that the vertex set of every level is same.

Compute Independent Spanners

- First we assume that the vertex set of every level is same.
- We run the greedy algorithm l times independently with stretch factors $t_1, t_2, \ldots, t_l$ where every time the input graph of the algorithm is G.

Compute Independent Spanners

- First we assume that the vertex set of every level is same.
- We run the greedy algorithm l times independently with stretch factors $t_1, t_2, \ldots, t_l$ where every time the input graph of the algorithm is G.
- If we assume $t_i = i$, then the total number of edges is $n^{1+\frac{1}{1}} + n^{1+\frac{1}{2}} + \cdots + n^{1+\frac{1}{l}} \leq n^{1+\frac{1}{2}}(1 + \frac{1}{n^{\frac{1}{6}}} + \frac{1}{n^{\frac{1}{4}}} + \frac{1}{n^{\frac{3}{10}}} + \cdots) \leq (1+\epsilon)n^{1+\frac{1}{2}}$, where $\epsilon \leq 2$.

Compute Independent Spanners

Lemma 6

There exists a graph having two independent spanners t_i-spanner and t_{i+1}-spanner such that t_{i+1}-spanner $\not\subseteq$ t_i-spanner.

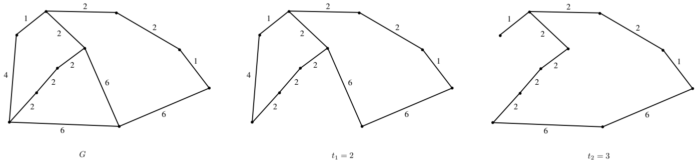


Figure: Independent inconsistent spanners

An Iterative Approach

- Generate spanner from G having stretch t_1.

An Iterative Approach

- Generate spanner from G having stretch t_1.
- To generate a t_2-spanner, instead of giving G as input we input the t_1-spanner we have computed in the previous step.

An Iterative Approach

Filtser et al. [4] have provided the following theorem that provides the relation between iterative spanners.

Lemma 7

Let $G = (V, E, w)$ be any weighted graph, let $t > 1$ be any stretch parameter, and let H be the greedy t-spanner of G. If H' is a t-spanner for H, then $H' = H$.

An Iterative Approach

- What happens if the stretch factor of two levels are different ($t_1 \neq t_2$)?

An Iterative Approach

- What happens if the stretch factor of two levels are different ($t_1 \neq t_2$)?
- We want to find the relationship between the spanner of second level and the input graph G (Is it a t_2-spanner of G?).

An Iterative Approach

Lemma 8

Let $G = (V, E, w)$ be any weighted graph, let $1 < t_1 < t_2$ be two stretch parameters, and let G_1 be the greedy t_1-spanner of G. If we compute a greedy spanner G_2 from G_1 with stretch parameter t_2, then G_2 may not be a t_2-spanner of G.

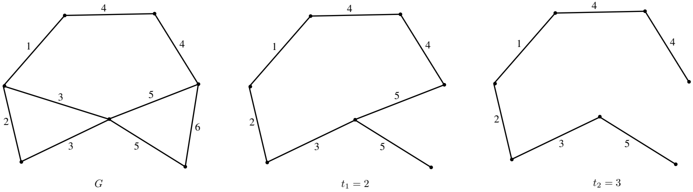


Figure: Spanners generated by iterative approach

An Iterative Approach

Lemma 9

Let $G = (V, E, w)$ be any weighted graph, let $1 < t_1 < t_2$ be two stretch parameters, and let G_1 be the greedy t_1-spanner of G. If we compute a greedy spanner G_2 from G_1 with stretch parameter t_2, then G_2 is a $(t_1 t_2)$-spanner of G.

An Iterative Approach

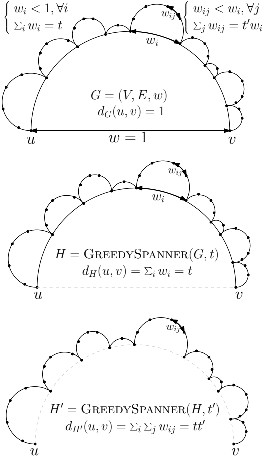


Figure: Spanners generated by iterative approach

Top-down and bottom-up approaches

Given: a nested set of vertices $V_1 \subset V_2 \subset \cdots \subset V_l$ where V_i is the terminal set of level l. We propose two algorithms to solve this general problem:

A Bottom-up Approach

- Decreasing stretch factors: $t_1 > t_2 > \cdots > t_l$ where t_i is the stretch factor of level i.

A Bottom-up Approach

- Decreasing stretch factors: $t_1 > t_2 > \cdots > t_l$ where t_i is the stretch factor of level i.
- Idea: find the spanner G_l of the bottom-most level.

A Bottom-up Approach

- Decreasing stretch factors: $t_1 > t_2 > \cdots > t_l$ where t_i is the stretch factor of level i.
- Idea: find the spanner G_l of the bottom-most level.
- G_l is a t_l-spanner of G and also a valid t_{l-1}-spanner, as $t_{l-1} > t_l$.

A Bottom-up Approach

- Problem: possibly extra edges in G_l. We may be able to remove some and still have a t_{l-1}-spanner.

A Bottom-up Approach

- Problem: possibly extra edges in G_l. We may be able to remove some and still have a t_{l-1}-spanner.
- Try to remove edges from G_l to get a spanner for level $l-1$. We repeat for the remaining spanners $G_{l-2}, G_{l-3}, \cdots, G_1$.

Bottom-up: Method

- Compute the t_l-spanner using the greedy algorithm. This spans all vertices.

Bottom-up: Method

- Compute the t_l-spanner using the greedy algorithm. This spans all vertices.
- For a t_{l-1} spanner, spanning V_{l-1}: divide edges in G_l into three types: neither vertex belongs to V_{l-1}, exactly one belongs, or both do.

Bottom-up: Method

- Consider every edge by descending order of weight.
- Remove it from the t_l-spanner and find all pair shortest paths. If there are two vertices in V_{l-1} such that the shortest path distance between them is more than t_{l-1} times the shortest path in G, then we put back the removed edge.
- Repeat for the second and third types of edges to get a t_{l-1}-spanner. Iteratively, we find spanners for each level on top.

A Top-down Approach

- Idea: ignore stretch factor by always taking the shortest paths.

A Top-down Approach

- Idea: ignore stretch factor by always taking the shortest paths.
- Find a t_1-spanner that spans the vertices in V_1.

A Top-down Approach

- Idea: ignore stretch factor by always taking the shortest paths.
- Find a t_1-spanner that spans the vertices in V_1.
- Use APSP to retain edges that appear in the shortest path between any pair of vertices in V_1.

A Top-down Approach

- Idea: ignore stretch factor by always taking the shortest paths.
- Find a t_1-spanner that spans the vertices in V_1.
- Use APSP to retain edges that appear in the shortest path between any pair of vertices in V_1.
- Repeat the procedure for V_2 by using the retained edges in the t_2-spanner, and so on.

Note on Steiner Tree Approximation

- The Steiner tree 2-approximation algorithm is similar. But it does not always give us a valid 1-spanner, since it finds the MST from the APSP.
- The APSP is always a valid 1-spanner. But after computing the MST, this is not true.

Note on Steiner Tree Approximation

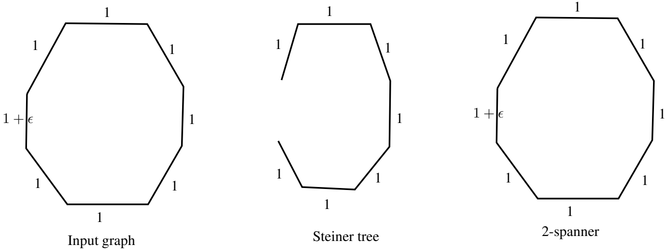


Figure: Here the Steiner tree is a 8-spanner but not a 2-spanner. And if we increase the number of vertices in the cycle, then we can show that the Steiner tree is not a valid t-spanner for any value of t. Note that here the edges represent a path, and the vertices represent terminals.

ILP Formulation

Definition 10 (Spanner Using ILP)

Given a connected, undirected graph $G = (V, E)$ with edge weights c_e, $e \in E$, a set of node pairs $K = \{(u_i, v_i)\}, i = 1, \cdots, k$, and a stretch factor $t > 1$:

Find $G' = (V, E')$ such that:

1. the size of $E' \in E$ is minimum
2. for every pair $(u_i, v_i) \in K$: the shortest path between (u_i, v_i) in G' is at most t times the shortest path in G.

If $K = V \times V$, then it is the spanner problem.

ILP Terminology

- P_{uv}: set of paths between u and v with length less than or equal to t times the shortest path in G, for all $(u, v) \in K$
- $P = \cup_{(u,v)\in K} P_{uv}$.
- δ_p^e: indicator variable stating whether the edge e is present in path p or not.
- x_e and y_p: decision variables, indicating whether the edge e and the path p is present in the spanner.

IP Formulation

$$\text{minimize} \sum_{e \in E} c_e x_e \tag{1}$$

Subject to:

$$\sum_{p \in P_{uv}} y_p \delta_p^e \leq x_e, \forall e \in E, \forall (u,v) \in K \tag{2}$$

$$\sum_{p \in P_{uv}} y_p \geq 1, \forall (u,v) \in K \tag{3}$$

$$x_e = \{0,1\}, \forall e \in E \tag{4}$$

$$y_p = \{0,1\}, \forall p \in P \tag{5}$$

Restricted Master Problem

- The number of variables may be exponential; thus, we solve the restricted master problem (RMP) instead:

Restricted Master Problem

- The number of variables may be exponential; thus, we solve the restricted master problem (RMP) instead:
-
$$\text{minimize} \sum_{e \in E} c_e x_e \tag{6}$$

Subject to:

$$\sum_{p \in P_{uv}} y_p \delta_p^e \leq x_e, \forall e \in E, \forall (u,v) \in K \tag{7}$$

$$\sum_{p \in P_{uv}} y_p \geq 1, \forall (u,v) \in K \tag{8}$$

$$x_e \geq 0, \forall e \in E \tag{9}$$

$$y_p \geq 0, \forall p \in P' \tag{10}$$

where $P'_{uv} \subseteq P_{uv}, P'_{uv} \neq \emptyset, \forall (u,v) \in K$ and $P' = \cup_{(u,v) \in K} P'_{uv}$.

Solving the LP

- Idea: delayed column generation
- Can start with only one path in P'_{uv}.
- Iteratively, add paths to P'.
- Now consider the dual LP with variables π_e^{uv} and $\sigma_{uv}, \forall e \in E, \forall (u, v) \in K$:

Dual LP Formulation

$$\text{maximize} \sum_{(u,v)\in K} \sigma_{uv} \tag{11}$$

Subject to:

$$\sum_{(u,v)\in K} \pi_e^{uv} \leq c_e, \forall e \in E \tag{12}$$

$$-\sum_{e\in E} \delta_p^e \pi_e^{uv} + \sigma_{uv} \leq 0, \forall p \in P_{uv}, \forall (u,v) \in K \tag{13}$$

$$\pi_e^{uv} \geq 0, \forall e \in E, \forall (u,v) \in K \tag{14}$$

$$\sigma_{uv} \geq 0, \forall (u,v) \in K \tag{15}$$

Solving the LP

- Solve the primal LP using Simplex method, and check if all constraints of the dual are satisfiable or not. If yes, we have found the optimal solution (weak duality theorem).

Solving the LP

- Solve the primal LP using Simplex method, and check if all constraints of the dual are satisfiable or not. If yes, we have found the optimal solution (weak duality theorem).
- Else we add a path in P' by solving the resource-constrained shortest path problem.

Solving the LP

- Solve the primal LP using Simplex method, and check if all constraints of the dual are satisfiable or not. If yes, we have found the optimal solution (weak duality theorem).
- Else we add a path in P' by solving the resource-constrained shortest path problem.
- For multi-level spanners, an edge selected at the ith level should also be selected in all lower levels.

Solving the LP

- Solve the primal LP using Simplex method, and check if all constraints of the dual are satisfiable or not. If yes, we have found the optimal solution (weak duality theorem).
- Else we add a path in P' by solving the resource-constrained shortest path problem.
- For multi-level spanners, an edge selected at the ith level should also be selected in all lower levels.
- Denote the selection of an edge e at level l by x_e^l where $x_e^l \in \{0, 1\}$. Add the constraint:

$$x_e^l \geq x_e^{l-1}, l \in \{2, 3, \ldots, k\} \tag{16}$$

Solving the LP

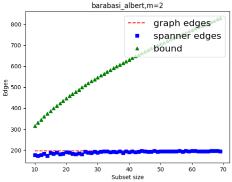
The primal LP remains the same for this formulation. We introduce another dual variable m_l^e, and the corresponding constraints become:

$$\sum_{(u,v)\in K} \pi_e^{uv} + m_l^e \leq c_e, \forall e \in E \tag{17}$$

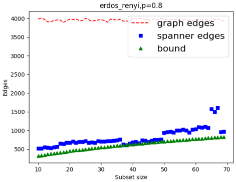
Experimental Results

- Test the algorithm for additive stretch 2 provided by Cygan et al. [2]
- Four graph generation models used
  - Erdös-Renyi, random geometric, Watts-Strogatz, Barabási-Albert
- Number of vertices $|V| = 100$
- Subset size $k = 10, \cdots, 100$

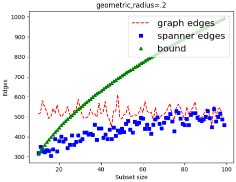
Plots



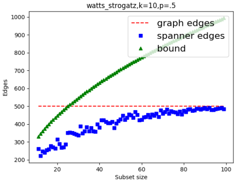
(a) BarabsiAlbert



(b) Erdős–Renyi



(c) Random Geometric



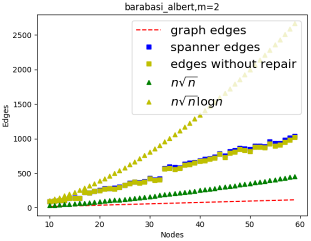
(d) Watts-Strogatz

Figure: Results for the single level spanner with additive stretch 2

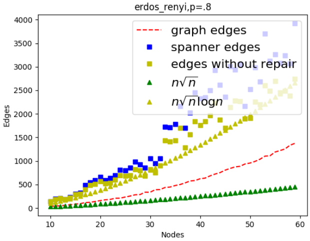
Experimental Results

- Consider the simple multi-level additive spanner
  - The size of the terminal sets increases exponentially
  - Every level has stretch 2
- Four graph generation models used
  - Erdös-Renyi, random geometric, Watts-Strogatz, Barabási-Albert
- Number of vertices $|V|$ up to 60

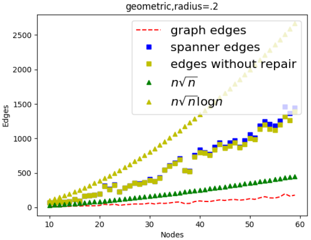
Plots

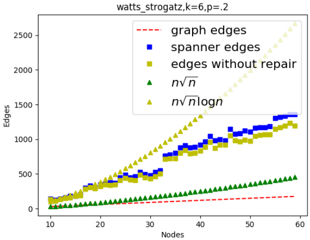


(a) BarabsiAlbert



(b) Erdős–Renyi





Plots

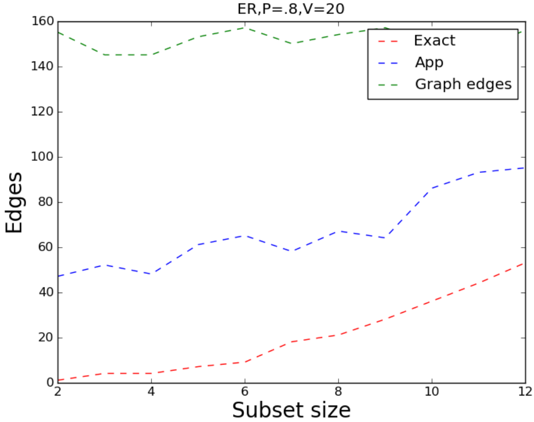


Figure: A comparison between the exact solution and the algorithm provided by Cygan et al. [2]

Future Work

- Implement the exact algorithm for multi-level.

Future Work

- Implement the exact algorithm for multi-level.
- Run experiment for large graphs.

Future Work

- Implement the exact algorithm for multi-level.
- Run experiment for large graphs.
- Consider the multiplicative version.

P. David and S. A. A., "Graph spanners," *Journal of Graph Theory*, vol. 13, no. 1, pp. 99–116.

M. Cygan, F. Grandoni, and T. Kavitha, "On Pairwise Spanners," in *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)* (N. Portier and T. Wilke, eds.), vol. 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (Dagstuhl, Germany), pp. 209–220, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013.

T. Kavitha, "New pairwise spanners," *Theory of Computing Systems*, vol. 61, pp. 1011–1036, Nov 2017.

A. Filtser and S. Solomon, "The greedy spanner is existentially optimal," in *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, (New York, NY, USA), pp. 9–17, ACM, 2016.