236610
Distributed Graph Algorithms

Spring 2018
Keren Censor-Hillel
Spanners

• Given $G=(V,E)$ and $E_S$ in $E$, a subgraph $S=(V,E_S)$ is called a **k-spanner** of $G$ if:
  
  – For every $u,v$ in $V$:
    
    $$\text{dist}_S(u,v) \leq k \cdot \text{dist}_G(u,v)$$
  
  – It is enough that the condition holds for every $u,v$ that are neighbors in $G$.

• $k$ is called the **stretch** of the spanner
Spanners

• Illustration

• Every graph is a 1-spanner of itself

• Why spanners?
  – Need a sparse subgraph
  – But sparsity increases distances
    • a tree may have a linear stretch
  – We care about the trade-off
Spanners

2-spanners:
• Clique?
• How sparse can it be in general?
  – $\Theta(n^2)$ edges in a complete bipartite graph
A (2k-1)-Spanner

• **Theorem:**
  Every graph has a \((2k-1)\)-spanner with \(O(n^{1+1/k})\) edges
A $(2k-1)$-Spanner

- **Theorem**: Every graph has a $(2k-1)$-spanner with $O(n^{1+1/k})$ edges

- **Proof**: By a greedy algorithm:
  - Initially, $S =$ empty
  - For every $e = \{u,v\}$ in $E$:
    - If $d_S(u,v) > (2k-1) \cdot d_G(u,v)$ then add $e$ to $S$
  - Return $S$
A (2k-1)-Spanner

- **Theorem:** Every graph has a **(2k-1)**-spanner with \(O(n^{1+1/k})\) edges

- **Proof:**
  - If \(d_S(u,v) > (2k-1)d_G(u,v)\) then add \(e\) to \(S\)

- **Stretch:** At most **2k-1**, otherwise \(e\) would have been added
A (2k-1)-Spanner

• **Theorem:**
  Every graph has a (2k-1)-spanner with $O(n^{1+1/k})$ edges

• **Proof:**
  – If $d_S(u,v) > (2k-1) d_G(u,v)$ then add $e$ to $S$

• **Edges:**

• **Claim 1:** $S$ has girth $\geq 2k+1$. 

Keren Censor-Hillel, Spring 2018
A (2k-1)-Spanner

- **Theorem:**
  Every graph has a (2k-1)-spanner with $O(n^{1+1/k})$ edges.

- **Proof:**
  - If $d_S(u,v) > (2k-1)d_G(u,v)$ then add $e$ to $S$

- **Edges:**

- **Claim 1: S has girth $\geq 2k+1$.**
  - If there was a cycle of length $2k$, consider last added edge.
A (2k-1)-Spanner

• **Theorem:**
  Every graph has a (2k-1)-spanner with $O(n^{1+1/k})$ edges

• **Proof:**
  – If $d_S(u,v) > (2k-1) d_G(u,v)$ then add $e$ to $S$

• **Edges:**

• **Claim 1:** $S$ has girth $\geq 2k+1$.
  – If there was a cycle of length $2k$, consider last added edge

• **Claim 2:** Any graph with girth $\geq 2k+1$ has $O(n^{1+1/k})$ edges.
A (2k-1)-Spanner

- **Claim 2:** Any graph with **girth** \( \geq 2k + 1 \) has \( O(n^{1+1/k}) \) edges.

- **Proof:** Suppose \( G \) with \( g \geq 2k + 1 \) and \( 10n^{1+1/k} \) edges.

  \( G' \): Loop and remove \( v \) with \( d(v) < 2n^{1/k} \).
  - \( g' \geq 2k+1 \) because subgraph of \( G \).
  - \( G' \) is not empty because removed at most \( 2n^{1+1/k} \) edges.
  - For all \( u \) in \( G' \): \( d(u) \geq 2n^{1/k} \)

  \( \rightarrow \) BFS from \( u \) of depth \( 2k \) has at least \( \sim(2n^{1/k})^k > n \) nodes
Girth conjecture (Erdös)

• **Conjecture:**
  There is a graph with girth $\geq 2k+2$ and $\Omega(n^{1+1/k})$ edges.

• Known for small values of $k$.

• If true, implies that the trade-off is optimal:
  – Remove any edge $\Rightarrow$ stretch at least $2k+1$
  – $\Rightarrow$ any $(2k-1)$-spanner = the entire graph
A Distributed (2k-1)-Spanner

• **Theorem**: There is a distributed algorithm that constructs a \((2k-1)\)-spanner with \(O(kn^{1+1/k})\) edges in \(O(k^2)\) rounds
Notation

• **Cluster**: A connected set of nodes \( C \) in \( V \)

• **Clustering**: A set of clusters \( P = \{C_1, \ldots, C_p\} \)

• Given a clustering \( P \), a node \( v \) is **covered** in \( P \) if there is a cluster \( C \) in \( P \) such that \( v \) is in \( C \).
  – We denote this cluster as \( C(v) \)

• A node \( v \) and a cluster \( C \) are called **neighbors** if there is a node \( u \) in \( C \) such that \( v \) and \( u \) are neighbors
Template

- $S =$ empty (spanner edges)
- Initially $P_0 = \{ \{ v \} \mid v \text{ in } V \}$

- For k-1 iterations:

- Given $P_{i-1}$, each $C$ in $P_{i-1}$ is **selected** with independent probability $\frac{1}{n^{1/k}}$
  - Denote $P'_i$ the set of selected clusters
Template

• For every node $v$ that is uncovered in $P'_i$

  – **Rule 1**: If $v$ has neighbors in $P'_i$ then $v$ joins one such neighbor $C$ and an edge from $v$ to $C$ is added to $S$

  – **Rule 2**: Otherwise, for every $C$ in $P_{i-1}$ that is a neighbor of $v$, an edge from $v$ to $C$ is added to $S$
Template

• The new clustering is $P_i$

  – If Rule 1 applied to $v$ then $v$ is covered in $P_i$

  – Otherwise, if Rule 2 applied to $v$ then $v$ is uncovered in $P_i$
Illustration

\[ C \text{ in } P_{i-1} \text{ selected to } P'_i \]

\[ C', C'' \text{ in } P_{i-1} \text{ not selected to } P'_i \]
Illustration

\( C \) in \( P_{i-1} \) selected to \( P'_i \)

\( C', C'' \) in \( P_{i-1} \) not selected to \( P'_i \)
Illustration

\( C \) in \( P_{i-1} \) selected to \( P'_i \)

\( C', C'' \) in \( P_{i-1} \) not selected to \( P'_i \)
Template

• Iteration $k$:
  
• $P_k$ is empty
  
  – Given $P_{k-1}$, each $C$ in $P_{k-1}$ is selected with probability 0

• **Rules 1 and 2** remain the same
Analysis – Number of Edges

• **Claim 1**: The expected number of edges in $S$ is $O(kn^{1+1/k})$

• **Proof**: We will see that the expected number of edges that are added to $S$ in each iteration is $O(n^{1+1/k})$

• Edges that are added according to **Rule 1** are at most one for each node, so their total number is at most $n$. 
Analysis – Number of Edges

• How many edges are added to $S$ according to Rule 2?

• Let $t$ be the number of clusters in $P_{i-1}$ that are neighbors of $v$.

• If $t \leq n^{1/k}$ then by Rule 2 we add at most $n^{1/k}$ edges to $S$
Analysis – Number of Edges

• Otherwise, denote $t = qn^{1/k}$, where $q > 1$.

• The probability for a cluster $C$ in $P_{i-1}$ to be selected into $P'_i$ is $1/n^{1/k}$

• So, the probability that no cluster in $P_{i-1}$ that is a neighbor of $v$ is selected is at most $(1 - 1/n^{1/k})^t$. 
Analysis – Number of Edges

• In this case we add \( t \) edges to \( S \) according to Rule 2. This gives that the expected number of edges that are added according to Rule 2 is at most:

\[
t(1 - \frac{1}{n^{1/k}})^t = qn^{1/k}((1 - \frac{1}{n^{1/k}})^n^{1/k})^q
\]

\[
= n^{1/k} q(1/e)^q
\]

This is < 1 for \( q > 1 \)

\[
= O(n^{1/k})
\]
Analysis – Number of Edges

• In iteration $k$:

• The probability of $C$ to survive all iterations is $(1/n^{1/k})^{k-1}$

• So for a node $v$, the number of edges added to $S$ in iteration $k$ is at most $n$ times the above, which is $n^{1/k}$. 
Analysis - Stretch

• **Claim 2**: The stretch of $S$ is at most $2k-1$

• **Proof**: Consider neighbors $v$ and $u$. We will see that $\text{dist}_S(u,v) \leq 2k-1$.

• Let $j$ be the minimal index such that either $u$ or $v$ is uncovered in $P_j$ (possibly both).
  – There must be such $j$ because in $P_0$ all nodes are covered and in $P_k$ none are covered.
Analysis - Stretch

• Assume w.l.o.g. that $u$ is uncovered. This means that Rule 2 was applied to $u$.

• Since $j$ is minimal, both $u$ and $v$ are covered in $P_{j-1}$.

• Since $u$ and $v$ are neighbors, there is an edge from $u$ to $C(v)$ that is added to $S$ according to Rule 2.
  – This may be an edge to some other $w$ in $C(v)$.
Illustration

$C(v)$ in $P_{j-1}$

Keren Censor-Hillel, Spring 2018
Analysis - Stretch

• This gives that:
\[
\text{dist}_S(u,v) \leq \text{dist}_S(u,w) + \text{dist}_S(w,v) \\
\leq 1 + 2(j-1) \\
\leq 2j-1 \\
\leq 2k-1
\]
Analysis - Stretch

• Why $\text{dist}_s(w,v) \leq 2(j-1)$ for $v$ and $w$ in the same $C$ in $P_{j-1}$?

• By induction on $j$, the radius of every $C$ in $P_j$ is at most $j$. That is, there is a $z$ in $C$ such that for every $y$ in $C$ we have $\text{dist}_s(z,y) \leq j$
Distributed Implementation

• Every component $C$, which was initially $\{z\}$, is maintained by its center $z$.
  – $z$ decides whether $C$ is selected in iteration $i$
  – Forwards this decision to all nodes of $C$

• Nodes of $C$ tell their neighbors whether $C$ is selected

• Every uncovered node $v$ knows whether to apply Rule 1 or Rule 2, and chooses edges accordingly
Distributed Implementation

• **Claim 3**: The distributed implementation completes in $O(k^2)$ rounds.

• **Proof**: In iteration $i$, it takes $i$ rounds for all nodes of a cluster $C$ to know whether it is selected or not (because $i$ is the radius of $C$).

• Another round is needed for telling the neighbors of $C$, and another round for uncovered nodes to respond.
Distributed Implementation

• This gives $O(i)$ rounds for iteration $i$
• The total number of rounds is:

$$\sum_{i=1}^{k} O(i) \leq O(k^2)$$

**Theorem:** There is a distributed algorithm that constructs a $(2k-1)$-spanner with $O(kn^{1+1/k})$ edges in $O(k^2)$ rounds
Additional Spanners

• We saw today a **multiplicative spanner**

• There are **(α, β)-spanners**, in which for every u and v in V:
  \[ \text{dist}_S(u,v) \leq \alpha \text{dist}_G(u,v) + \beta \]
  – It is no longer enough that the condition holds for neighbors in G

• There are **purely additive c-spanners**, in which for every u and v in V:
  \[ \text{dist}_S(u,v) \leq \text{dist}_G(u,v) + c \]