The Sparsest Additive Spanner via Multiple Weighted BFS Trees

Ami Paz IRIF–CNRS & Paris Diderot University
Joint work with:
Keren Censor-Hillel, Noam Ravid Technion

This project has received funding from the European Union's Horizon 2020 Research and Innovation Program under grant agreement no. 755839
The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous
The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous
- Input
  - Unique ID
  - Neighbors
The CONGEST Model

- Communication graph on \( |V| = n \) nodes
- Bounded messages, \( O(\log n) \) bits
- Synchronous
- Input
  - Unique ID
  - Neighbors
- Output
  - Neighbors in the spanner
The CONGEST Model

- Communication graph on $|V| = n$ nodes
- Bounded messages, $O(\log n)$ bits
- Synchronous
- Input
  - Unique ID
  - Neighbors
- Output
  - Neighbors in the spanner
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS
Example: Distributed BFS

- BFS in $O(D)$ rounds
Example: Multiple BFS trees
Example: Multiple BFS trees
Example: Multiple BFS trees
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source

Message format: $(\text{dist, source})$
Example: Multiple BFS trees

- Prioritize by distance
- Secondary: by source

Message format: $(\text{dist}, \text{source})$
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source
- Here:
  - $s_1$ before $s_2$

Message format: $(dist, source)$
Example: Multiple BFS trees

- Prioritize by \textit{distance}
  - Secondary: by source

Message format: \((\text{dist, source})\)
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source

\[ s_1 \]
\[ s_2 \]
Example: Multiple BFS trees

- Prioritize by distance
  - Secondary: by source
Example: Multiple BFS trees

BFS from $\tau$ sources

- Trivial: $O(\tau \cdot D)$ rounds

**Theorem [LP13]**

It is possible to construct BFS trees from $\tau$ sources in $O(\tau + D)$ rounds.
Weighted BFS

$G$ weighted graph, $\tau$ source

Want: a BFS tree with minimal-weight paths from $s$

That is: from all shortest $(s, t)$-paths, find the lightest
Weighted BFS

\( G \) weighted graph, \( \tau \) source

Want: a BFS tree with minimal-weight paths from \( s \)

• Is this a tree?
• Can we build it in CONGEST?
• Can we build multiple trees?
Weighted BFS

Claims:

• There is a tree with shortest-lightest paths
• It can be built in CONGEST
Weighted BFS

Claims:

• There is a tree with shortest-lightest paths
• It can be built in CONGEST

Message format:

\((dist, source, w\_dist)\)
Weighted BFS

Claims:

• There is a tree with shortest-lightest paths
• It can be built in CONGEST
Weighted BFS

\( G \) weighted graph, \( s \) source

Want: a BFS tree with minimal weight paths from \( s \)

• Is this a tree? 😊
• Can we build it in CONGEST? 😊
• Can we build multiple trees?
Weighted BFS

Claim:

• We can build multiple wBFS trees in CONGEST

Message format: 
(dist, source, w_dist)
Weighted BFS

Claim:

• We can build multiple wBFS trees in CONGEST

Message format: \((dist, source, w\_dist)\)
Weighted BFS

Claim:

- We can be build multiple wBFS trees in CONGEST

Message format:
$(dist, source, w\_dist)$
Weighted BFS

Claim:

- We can build multiple wBFS trees in CONGEST

Message format: 
$(dist, source, w_{dist})$
Weighted BFS

Claim:

- We can build multiple wBFS trees in CONGEST

Message format: $(\text{dist}, \text{source}, \text{w}_\text{dist})$

Sends $s_1$ first
Weighted BFS

Claim:

- We can build multiple wBFS trees in CONGEST

Message format: $(\text{dist}, \text{source}, \text{w\_dist})$

All updates arrive before $t$ sends (nontrivial)
Weighted BFS

\( G \) weighted graph, \( s \) source

Want: a BFS tree with minimal weight paths from \( s \)

- Is this a tree? 😊
- Can we build it in CONGEST? 😊
- Can we build multiple trees? 😊
Multiple Weighted BFS trees

Weighted BFS from $\tau$ sources

**Theorem (New)**

It is possible to construct weighted BFS trees from $\tau$ sources in $O(\tau + D)$ rounds
Spanners

A graph $G$ on $n$ nodes

Want: a subgraph $H$ on the same nodes, that

• Approximately preserves distances
• Sparse
Spanners

A graph $G$ on $n$ nodes

Want: a subgraph $H$ on the same nodes, that

- Approximately preserves distances
- Sparse

This talk:

only additive all-pairs spanners
Spanners

A \((+\beta)\)-spanner of \(G\) is a subgraph \(H\) on the same nodes, s.t.

- for all \((u, v) \in V \times V:\)
  \[
  \text{dist}_H(u, v) \leq \text{dist}_G(u, v) + \beta
  \]
Spanners

A \((+\beta)\)-spanner of \(G\) is a subgraph \(H\) on the same nodes, s.t.

- for all \((u, v) \in V \times V:\n\text{dist}_H(u, v) \leq \text{dist}_G(u, v) + \beta
Spanners

A \((+\beta)\)-spanner of \(G\) is a subgraph \(H\) on the same nodes, s.t.

- for all \((u, v) \in V \times V\):
  \[
  \text{dist}_H(u, v) \leq \text{dist}_G(u, v) + \beta
  \]
Spanners

A \((+\beta)\)-spanner of \(G\) is a subgraph \(H\) on the same nodes, s.t.

- for all \((u, v) \in V \times V:\)
  \[\text{dist}_H(u, v) \leq \text{dist}_G(u, v) + \beta\]
Applications

• Synchronizers [Awe85,PU89]
• Information dissemination [CHHKM12]
• Compact routing schemes [PU89,TZ01,Che13]
• And many more...
Sequential Spanners

• Constructions
  • (+2): $O(n^{3/2})$ edges [ACIM99]
  • (+4): $\tilde{O}(n^{7/5})$ edges [Che13]
  • (+6): $O(n^{4/3})$ edges [BKMP10]

• Lower bound
  • Any: $n^{4/3}/2^{\Omega(\sqrt{\log n})}$ edges [AB16]

Goal:
Networks that build their own spanners
# Distributed Additive Spanners

<table>
<thead>
<tr>
<th>Spanner</th>
<th>Number of edges</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+2)-spanner</td>
<td>$O(n^{3/2})$</td>
<td>[ACIM99]</td>
</tr>
<tr>
<td>(+4)-spanner</td>
<td>$\tilde{O}(n^{7/5})$</td>
<td>[Che13]</td>
</tr>
<tr>
<td>(+6)-spanner</td>
<td>$O(n^{4/3})$</td>
<td>[BKMP10]</td>
</tr>
<tr>
<td>(+8)-spanner</td>
<td>Optimal</td>
<td></td>
</tr>
<tr>
<td>(+?)-spanner</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Distributed Additive Spanners

<table>
<thead>
<tr>
<th>Spanner</th>
<th>Number of edges</th>
<th>Sequential</th>
<th>Distributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+2)-spanner</td>
<td>(O(n^{3/2})) [ACIM99]</td>
<td>(\tilde{O}(n^{3/2})) [LP13]</td>
<td></td>
</tr>
<tr>
<td>(+4)-spanner</td>
<td>(\tilde{O}(n^{7/5})) [Che13]</td>
<td>(\tilde{O}(n^{7/5})) [CH+17]</td>
<td></td>
</tr>
<tr>
<td>(+6)-spanner</td>
<td>(O(n^{4/3})) [BKMP10]</td>
<td>(\tilde{O}(n^{15/11})) [CH+17]</td>
<td></td>
</tr>
<tr>
<td>(+8)-spanner</td>
<td>(\tilde{O}(n^{4/3})) (???)</td>
<td>(O(n^{4/3})) (???)</td>
<td></td>
</tr>
</tbody>
</table>
## Distributed Additive Spanners

<table>
<thead>
<tr>
<th>Spanner</th>
<th>Number of edges</th>
<th>Sequential</th>
<th>Distributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+2)-spanner</td>
<td>$O(n^{3/2})$ [ACIM99]</td>
<td>$\tilde{O}(n^{3/2})$ [LP13]</td>
<td></td>
</tr>
<tr>
<td>(+4)-spanner</td>
<td>$\tilde{O}(n^{7/5})$ [Che13]</td>
<td>$\tilde{O}(n^{7/5})$ [CH+17]</td>
<td></td>
</tr>
<tr>
<td>(+6)-spanner</td>
<td>$O(n^{4/3})$ [BKMP10]</td>
<td>$\tilde{O}(n^{4/3})$</td>
<td></td>
</tr>
<tr>
<td>(+8)-spanner</td>
<td>$\tilde{O}(n^{15/11})$ [CH+17]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(+?)-spanner</td>
<td>$O(n^{4/3})$ (???)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Spanner Construction

Two phases:

• Clustering

• Path buying
Clustering

• Choose nodes as centers \textit{at random}

• \textbf{Add edges} to their neighbors
  • All \textit{high-degree nodes} are clustered w.h.p.

• Add all edges of \textit{un-clustered} nodes
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
  - All high-degree nodes are clustered w.h.p.
- Add all edges of un-clustered nodes
Clustering

- Choose nodes as centers \textit{at random}
- \textbf{Add edges} to their neighbors
  - All \textit{high-degree nodes} are clustered w.h.p.
- \textbf{Add all edges of} \textit{un-clustered} nodes
Clustering

• Choose nodes as centers at random
• Add edges to their neighbors
  • All high-degree nodes are clustered w.h.p.
• Add all edges of un-clustered nodes
Clustering

• Choose nodes as centers at random
• Add edges to their neighbors
  • All high-degree nodes are clustered w.h.p.
• Add all edges of un-clustered nodes
Clustering

• Choose nodes as centers at random
• Add edges to their neighbors
  • All high-degree nodes are clustered w.h.p.
• Add all edges of un-clustered nodes
Clustering

• Choose nodes as centers at random

• Add edges to their neighbors
  • All high-degree nodes are clustered w.h.p.

• Add all edges of un-clustered nodes
Path Buying

• For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:
  • Build a set $S_k$ of $\sim 1/k$ of the clusters
  • For each center $c_i$ and a cluster $C_j \in S_k$
    • Add a shortest path from $c_i$ to some $v \in C_j$
    • But only if it misses at most $2k$ edges
Path Buying

For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:

- Build a set $S_k$ of $\sim 1/k$ of the clusters
- For each center $c_i$ and a cluster $C_j \in S_k$
  - Add a shortest path from $c_i$ to some $v \in C_j$
  - But only if it misses at most $2k$ edges

That is, for each $(c_i, C_j)$:

1. $A \leftarrow \emptyset$
2. For each $v \in C_j$, if there is a $(c_i, v)$-path that is shortest and misses $\leq 2k$ edges add one to $A$
3. If $A \neq \emptyset$, add a shortest path from $A$
Path Buying

• For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:
  • Build a set $S_k$ of $\sim 1/k$ of the clusters
  • For each center $c_i$ and a cluster $C_j \in S_k$
    • Add a shortest path from $c_i$ to some $v \in C_j$
    • But only if it misses at most $2k$ edges
Path Buying

• For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:
  • Build a set $S_k$ of $\sim 1/k$ of the clusters
  • For each center $c_i$ and a cluster $C_j \in S_k$
    • Add a shortest path from $c_i$ to some $v \in C_j$
    • But only if it misses at most $2k$ edges
Path Buying

• For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:
  • Build a set $S_k$ of $\sim 1/k$ of the clusters
  • For each center $c_i$ and a cluster $C_j \in S_k$
    • Add a shortest path from $c_i$ to some $v \in C_j$
    • But only if it misses at most $2k$ edges
Path Buying

• For \( k = 1, 2, 4, 8, \ldots, \frac{n^2}{3} \) do:
  • Build a set \( S_k \) of \( \sim \frac{1}{k} \) of the clusters
  • For each center \( c_i \) and a cluster \( C_j \in S_k \)
    • Add a shortest path from \( c_i \) to some \( v \in C_j \)
    • But only if it misses at most \( 2k \) edges
Path Buying

• For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:
  • Build a set $S_k$ of $\sim 1/k$ of the clusters
  • For each center $c_i$ and a cluster $C_j \in S_k$
    • Add a shortest path from $c_i$ to some $v \in C_j$
    • But only if it misses at most $2k$ edges
Theorem (New)

It is possible to construct:

- A \((+6)\)-spanner
- With \(\tilde{O}(n^{4/3})\) edges
- In \(\tilde{O}(n^{2/3} + D)\) rounds
Clustering

- Choose nodes as centers at random
- Add edges to their neighbors
- Add all edges of un-clustered nodes

Locally

Talk to neighbors

Talk to neighbors
Path Buying

• For $k = 1, 2, 4, 8, \ldots, n^{2/3}$ do:
  • Build a set $S_k$ of $\sim 1/k$ of the clusters
  • For each center $c_i$ and a cluster $C_j \in S_k$
    • Add a shortest path from $c_i$ to some $v \in C_j$
    • But only if it misses at most $2k$ edges

Join locally to $S_k$

For each $(c_i, C_j)$, for each $v \in C_j$, need to find the shortest $(c_i, v)$-path that misses minimal num. of edges

Note: Graph and spanner are unweighted
Only use weights for the alg.

Weight edges: missing=1, others=0
Run wBFS from each $c_i$
Distributed Spanner Construction

**Theorem (New)**

It is possible to construct:

- A (+6)-spanner
- With $\tilde{O}(n^{4/3})$ edges
- In $\tilde{O}(n^{2/3} + D)$ rounds
Conclusion

• New sequential algorithm for (±6)-spanners
• New distributed implementation
  • Gives an almost-optimal (±6)-spanner
• New distributed algorithm: weighted-BFS

• Open: lower bounds for distributed construction time

Thank You!