Pairwise Additive Spanners

Nithin M. Varma
(Joint work with T. Kavitha)

TIFR Mumbai → Penn State

Theory Seminar

September 22, 2014

Appeared in part in the proceedings of ICALP 2013

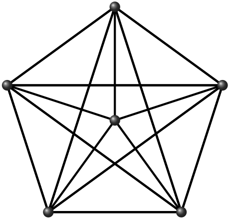
Graph Spanners

Peleg and Schaffer 1989

$H = (V, E')$ is a spanner of $G = (V, E)$, an undirected unweighted graph, if

- H is a subgraph of G ($E' \subseteq E$)
- $d_H(u, v) \approx d_G(u, v)$ for all $u, v \in V(G)$

$G = K_6$

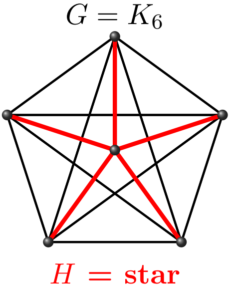


Graph Spanners

Peleg and Schaffer 1989

$H = (V, E')$ is a spanner of $G = (V, E)$, an undirected unweighted graph, if

- H is a subgraph of G ($E' \subseteq E$)
- $d_H(u, v) \approx d_G(u, v)$ for all $u, v \in V(G)$



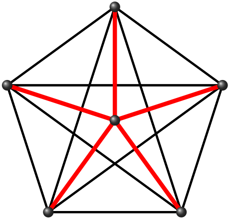
Graph Spanners

Peleg and Schaffer 1989

$H = (V, E')$ is a spanner of $G = (V, E)$, an undirected unweighted graph, if

- H is a subgraph of G ($E' \subseteq E$)
- $d_H(u, v) \approx d_G(u, v)$ for all $u, v \in V(G)$

$G = K_6$



H = star

For all $u, v \in V$

- $d_H(u, v) \leq 2 \cdot d_G(u, v)$ (multiplicative)

- $\boxed{d_H(u, v) \leq d_G(u, v) + 1 \text{ (additive)}}$

Why Spanners?

- Fewer edges than the original graph, but roughly preserve shortest distances

Why Spanners?

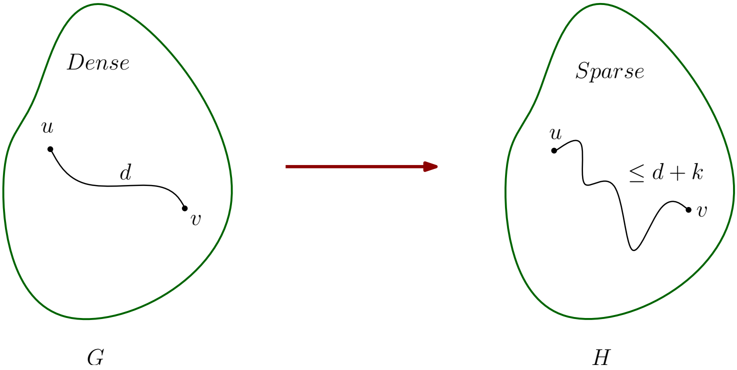
- Fewer edges than the original graph, but roughly preserve shortest distances
- Algorithms run on spanners can give approximate solutions for problems on original graph

Why Spanners?

- Fewer edges than the original graph, but roughly preserve shortest distances
- Algorithms run on spanners can give approximate solutions for problems on original graph

- Space Efficient Routing Schemes
  - Thorup and Zwick (2001)
- Near Shortest Path Algorithms
  - Elkin (2001)
- Approximate Distance Oracles
  - Patrascu and Roditty (2010)

Additive Spanners



Liestman and Shermer (1991)

H is a $+k$-spanner of G if $d_H(u, v) \leq d_G(u, v) + k$ for all $u, v \in V$.

Bounds for Additive Spanners

Upper Bounds

- $+2$-spanner with $O(n^{1.5})$ edges
  - Dor, Halperin and Zwick (2000, $\tilde{O}(n^{1.5})$ edges)
  - Elkin and Peleg (2001, $O(n^{1.5})$ edges)
- $+4$-spanner with $\tilde{O}(n^{1.4})$ edges
  - Chechik (2013)
- $+6$-spanner with $O(n^{1.33})$ edges
  - Baswana, Kavitha, Mehlhorn, Pettie (2005)
- $+\tilde{O}(n^{\frac{1-3\delta}{2}})$-spanner with $\tilde{O}(n^{1+\delta})$ edges for $\delta \in [\frac{3}{17}, \frac{1}{3})$
  - Chechik (2013)

Bounds for Additive Spanners

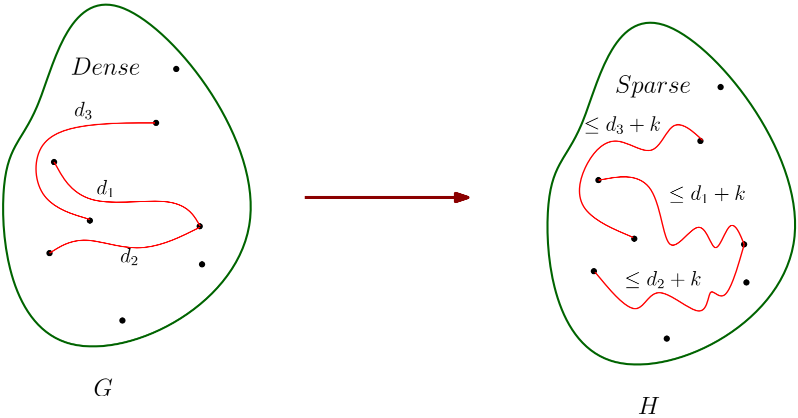
Upper Bounds

- $+2$-spanner with $O(n^{1.5})$ edges
  - Dor, Halperin and Zwick (2000, $\tilde{O}(n^{1.5})$ edges)
  - Elkin and Peleg (2001, $O(n^{1.5})$ edges)
- $+4$-spanner with $\tilde{O}(n^{1.4})$ edges
  - Chechik (2013)
- $+6$-spanner with $O(n^{1.33})$ edges
  - Baswana, Kavitha, Mehlhorn, Pettie (2005)
- $+\tilde{O}(n^{\frac{1-3\delta}{2}})$-spanner with $\tilde{O}(n^{1+\delta})$ edges for $\delta \in [\frac{3}{17}, \frac{1}{3})$
  - Chechik (2013)

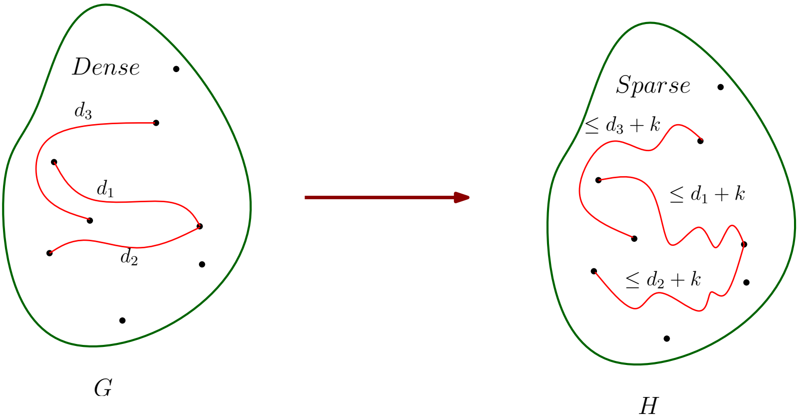
Lower Bounds

- $\Omega(n^{1+\frac{1}{k}})$ edges necessary, for $+(2k-1)$-spanners
  - Woodruff (2006)

Our Focus: Pairwise Additive Spanners



Our Focus: Pairwise Additive Spanners



Cygan, Grandoni, Kavitha(2013)

A generalization of spanners: not all pairs in $V \times V$ are important here, only certain pairs are critical.

Pairwise Additive Spanners : Two Variants

$\mathcal{P}$**-spanners**[Cygan, Grandoni, Kavitha (2013)]

- Set of pairs explicitly given as $\mathcal{P} \subseteq V \times V$.

D**-spanners**[Kavitha, V. (2013)]

- Set of pairs specified implicitly using a number D
- $\mathcal{P} = \{(u, v) : d(u, v) \geq D\}$

Our Results [Kavitha, V. (2013)]

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

Our Results [Kavitha, V. (2013)]

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$
- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

Our Results [Kavitha, V. (2013)]

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$
- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

D-spanners

- $+4k$ D-spanner with $\tilde{O}(n^{1.5}/D^{k/(2k+2)})$ edges for any integer $k \geq 1$

Our Results [Kavitha, V. (2013)]

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

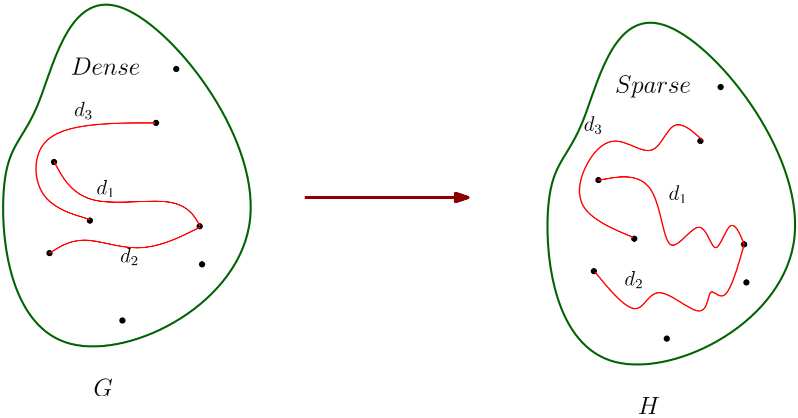
$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$
- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

D-spanners

- $+4k$ D-spanner with $\tilde{O}(n^{1.5}/D^{k/(2k+2)})$ edges for any integer $k \geq 1$
    - $+4$ D-spanner with $\tilde{O}(n^{1.5}/D^{0.25})$ edges
    - $+4\log n$ D-spanner with $\tilde{O}(n\sqrt{n/D})$ edges

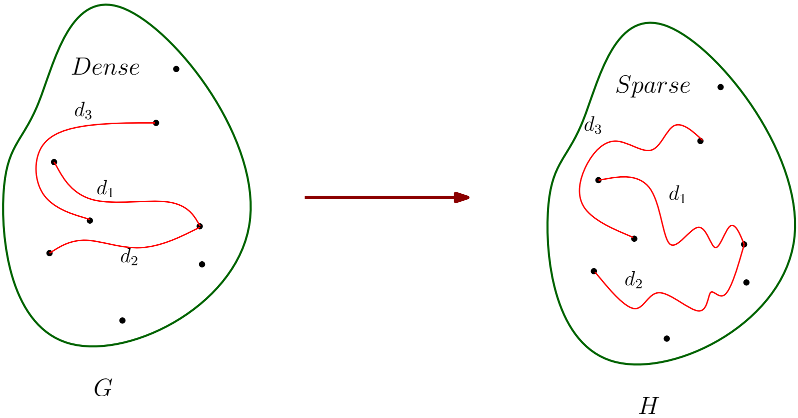
$\mathcal{P}$-preservers



Coppersmith and Elkin (2006)

H is a $\mathcal{P}$-preserver of G if $d_H(u, v) = d_G(u, v)$ whenever $(u, v) \in \mathcal{P}$, where $\mathcal{P} \subseteq V \times V$.

D-preservers



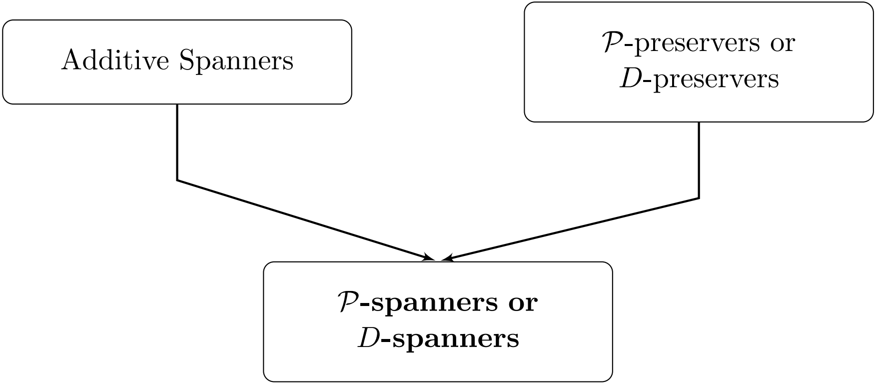
Bollobás, Coppersmith and Elkin (2005)

H is a D-preserver of G if $d_H(u,v) = d_G(u,v)$ whenever $d_G(u,v) \geq D$

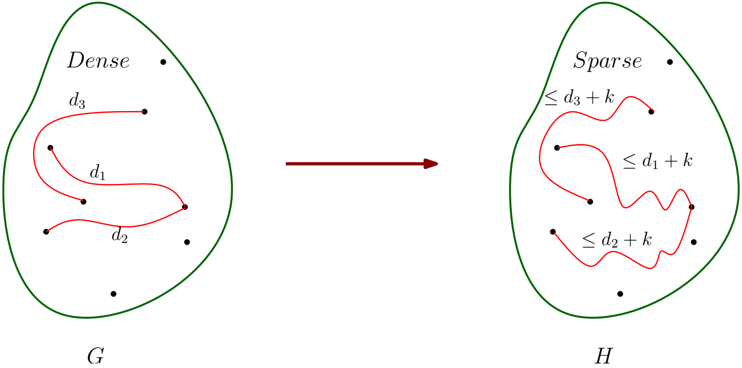
Bounds

- D-preserver with $O(n^2/D)$ edges (This is tight.)
  - Bollobás, Coppersmith and Elkin (2005)
- $\mathcal{P}$-preserver with $O(\min(n\sqrt{|\mathcal{P}|}, |\mathcal{P}|\sqrt{n}))$ edges
  - Coppersmith and Elkin (2006)

Today


Additive Spanners
$\mathcal{P}$-preservers or
D-preservers
$\mathcal{P}$-spanners or
D-spanners

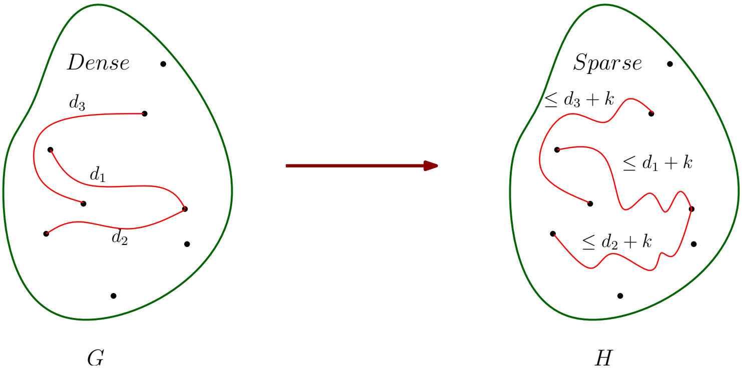
$\mathcal{P}$-spanners



Cygan, Grandoni, Kavitha 2013

H is a $+k$ $\mathcal{P}$-spanner of G if $d_H(u,v) \leq d_G(u,v) + k$ whenever $(u,v) \in \mathcal{P}$, where $\mathcal{P} \subseteq V \times V$.

D-spanners



Kavitha, V. 2013

H is a $+k$ D-spanner of G if $d_H(u,v) \leq d_G(u,v) + k$ whenever $d_G(u,v) \geq D$.

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

D-spanners

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

D-spanners

- $+4k$ D-spanner with $\tilde{O}(n^{1.5}/D^{k/(2k+2)})$ edges for any integer $k \geq 1$

Our Results

There exist deterministic polynomial time algorithms, which given any graph $G = (V, E)$ on n vertices, construct:

$\mathcal{P}$-spanners

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges for any $\mathcal{P} \subseteq V \times V$

- A $+2$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/4})$ edges when $\mathcal{P} = S \times V$ for any $S \subseteq V$
- A $+4$ $\mathcal{P}$-spanner with $\tilde{O}(n|\mathcal{P}|^{1/5}) = \tilde{O}(n^{1.4})$ edges when $\mathcal{P} = V \times V$

D-spanners

- $+4k$ D-spanner with $\tilde{O}(n^{1.5}/D^{k/(2k+2)})$ edges for any integer $k \geq 1$
  - $+4$ D-spanner with $\tilde{O}(n^{1.5}/D^{0.25})$ edges
  - $+4\log n$ D-spanner with $\tilde{O}(n\sqrt{n/D})$ edges

What we are going to prove..

Theorem

There is a polynomial time algorithm which, given any graph $G = (V, E)$ on n nodes and any $\mathcal{P} \subseteq V \times V$, computes a $+2$ $\mathcal{P}$-spanner of G with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges.

+2 $\mathcal{P}$-spanner algorithm

Input

- Graph $G = (V, E)$ on n vertices
- Set $\mathcal{P} \subseteq V \times V$ of pairs to be approximated

Output

- $H = (V, E')$
- H is a $+2$ $\mathcal{P}$-spanner of G
- H has $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges

Main Algorithmic Techniques Used

- Clustering
  - [EP01,BKMP05,C13,CGK13]

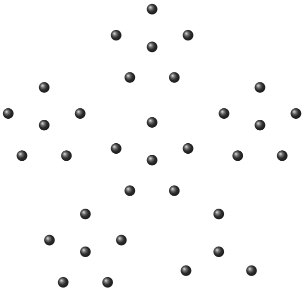
Main Algorithmic Techniques Used

- Clustering
  - [EP01,BKMP05,C13,CGK13]
- Shortest Paths Tree Addition
  - [EP01,C13]

Main Algorithmic Techniques Used

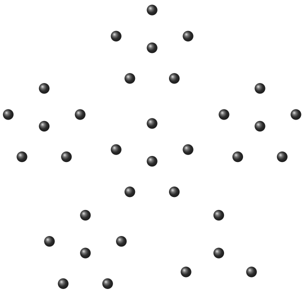
- Clustering
  - [EP01,BKMP05,C13,CGK13]
- Shortest Paths Tree Addition
  - [EP01,C13]
- Path Buying
  - [BKMP05,C13,CGK13]

Construction : Initialization



- Initialize H to the empty graph.

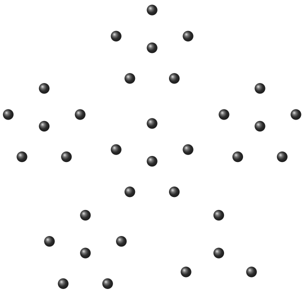
Construction : Clustering



Forming Clusters

- Mark all nodes as **unclustered**

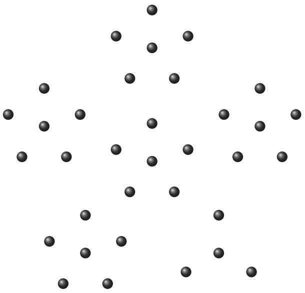
Construction : Clustering



Forming Clusters

- Mark all nodes as **unclustered**
- Repeat the following steps.

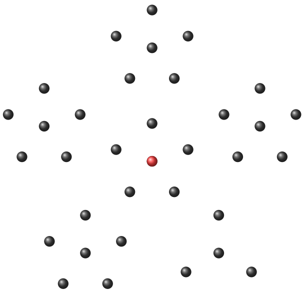
Construction : Clustering



Forming Clusters

- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.

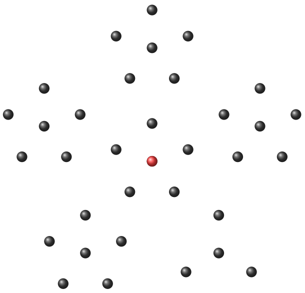
Construction : Clustering



Forming Clusters

- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.

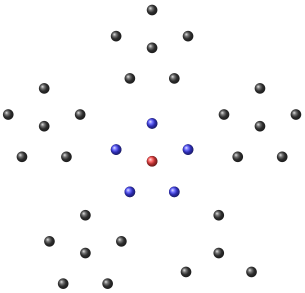
Construction : Clustering



Forming Clusters

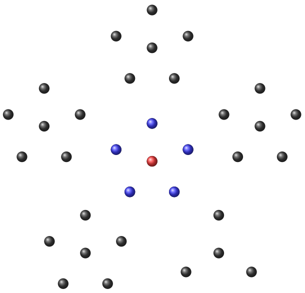
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.

Construction : Clustering



Forming Clusters

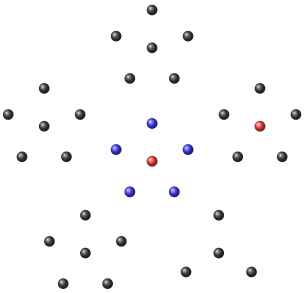
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.



Forming Clusters

- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers

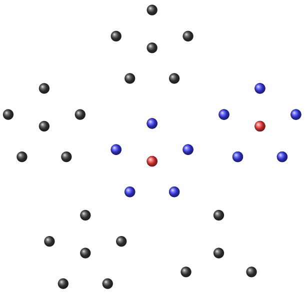
Construction : Clustering



Forming Clusters

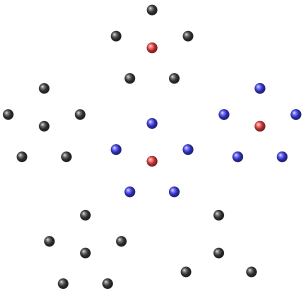
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers

Construction : Clustering



Forming Clusters

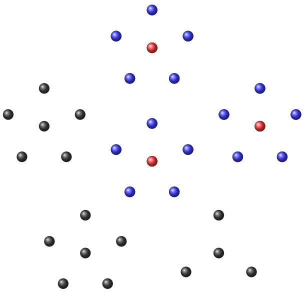
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers



Forming Clusters

- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers

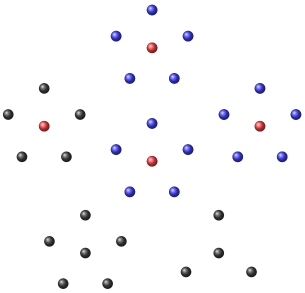
Construction : Clustering



Forming Clusters

- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers

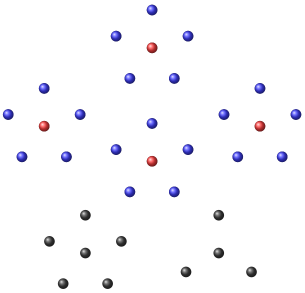
Construction : Clustering



Forming Clusters

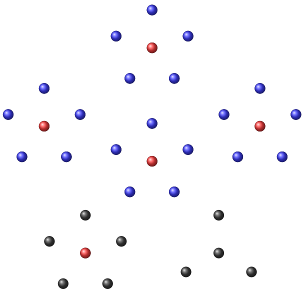
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers

Construction : Clustering



Forming Clusters

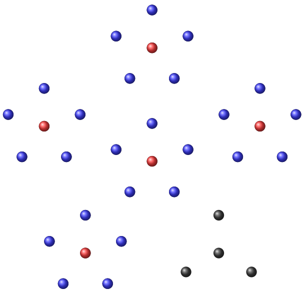
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers



Forming Clusters

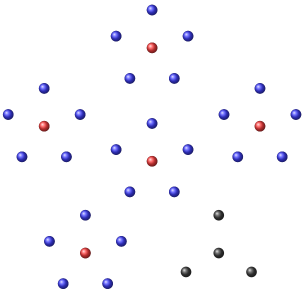
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers

Construction : Clustering



Forming Clusters

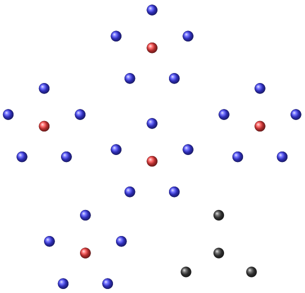
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers



Forming Clusters

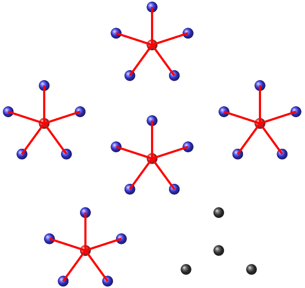
- Mark all nodes as **unclustered**
- Repeat the following steps.
- Mark a node with at least h $(= (|\mathcal{P}| \cdot \log n)^{1/3})$ **unclustered** neighbors as a cluster center.
- Mark all its **unclustered** neighbors as clustered and form a cluster.
- Stop when there are no potential cluster centers
- Some nodes remain **unclustered**.

Construction : Clustering



Adding edges

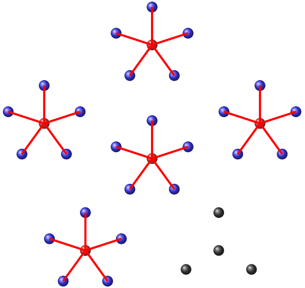
Construction : Clustering



Adding edges

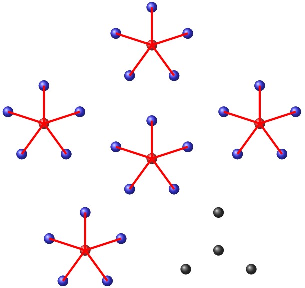
- Add the edges between cluster centers and nodes in their cluster to H.

Construction : Clustering



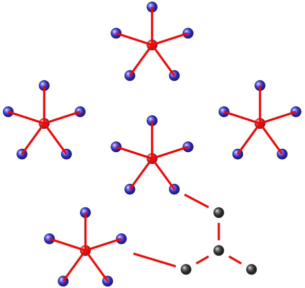
Adding edges

- Add the edges between cluster centers and nodes in their cluster to H.



Adding edges

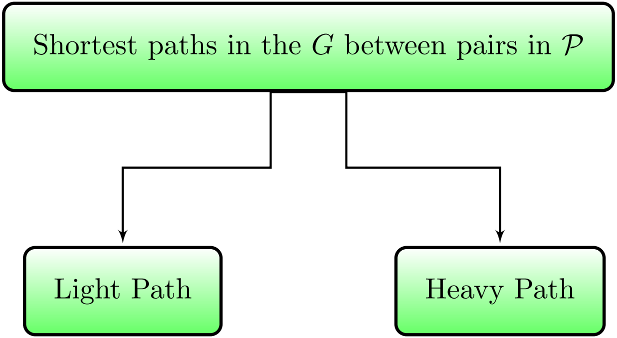
- Add the edges between cluster centers and nodes in their cluster to H.
- Add all the edges incident on **unclustered** nodes to H.



Adding edges

- Add the edges between cluster centers and nodes in their cluster to H.
- Add all the edges incident on **unclustered** nodes to H.

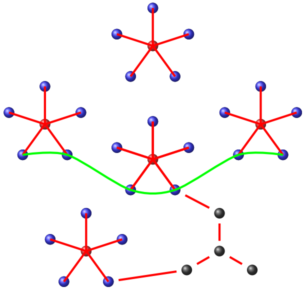
After Clustering



$< n \log n / h^2$
clustered nodes

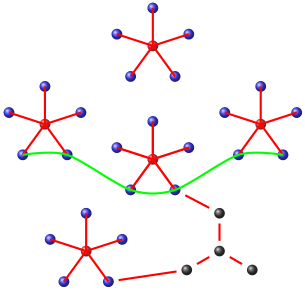
$\geq n \log n / h^2$
clustered nodes

Construction : Shortest Paths Tree Addition



A heavy path intersects *many* ($\geq n \log n / 3h^2$) clusters

Construction : Shortest Paths Tree Addition

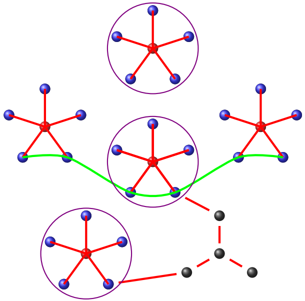


A heavy path intersects *many* ($\geq n \log n / 3h^2$) clusters

$$\Downarrow$$

$\exists$ a collection of $O(h)$ clusters $\mathcal{C}$ such that every heavy path intersects some cluster in $\mathcal{C}$.

Construction : Shortest Paths Tree Addition

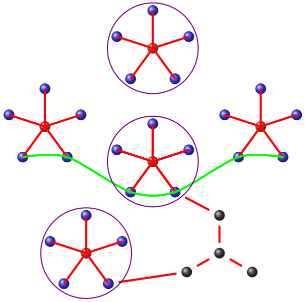


A heavy path intersects *many* ($\geq n \log n / 3h^2$) clusters

$$\Downarrow$$

$\exists$ a collection of $O(h)$ clusters $\mathcal{C}$ such that every heavy path intersects some cluster in $\mathcal{C}$.

Construction : Shortest Paths Tree Addition



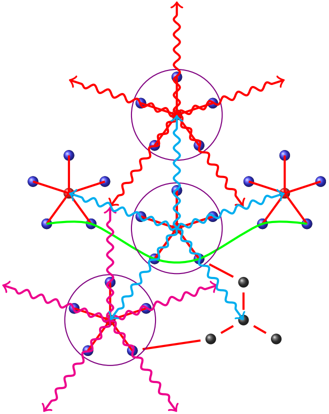
A heavy path intersects *many* ($\geq n \log n / 3h^2$) clusters

$$\Downarrow$$

$\exists$ a collection of $O(h)$ clusters $\mathcal{C}$ such that every heavy path intersects some cluster in $\mathcal{C}$.

- Add the edges in the union of SPTs in G rooted at these cluster centers to H.

Construction : Shortest Paths Tree Addition



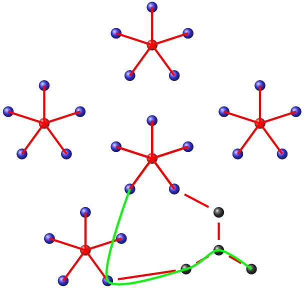
A heavy path intersects *many* ($\geq n \log n / 3h^2$) clusters

$$\Downarrow$$

$\exists$ a collection of $O(h)$ clusters $\mathcal{C}$ such that every heavy path intersects some cluster in $\mathcal{C}$.

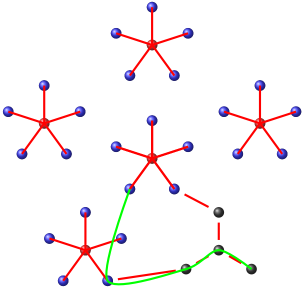
- Add the edges in the union of SPTs in G rooted at these cluster centers to H.

Construction : Path Buying



A light path has a *few* ($< n \log n / h^2$) clustered nodes

Construction : Path Buying

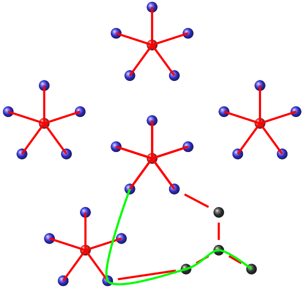


A light path has a *few* ($< n \log n/h^2$) clustered nodes

$$\Downarrow$$

Except a *few* ($< n \log n/h^2$) edges, most of the edges are already present in H.

Construction : Path Buying

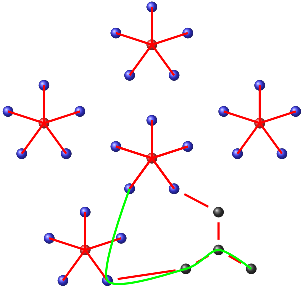


A light path has a *few* ($< n \log n/h^2$) clustered nodes

$$\Downarrow$$

Except a *few* ($< n \log n/h^2$) edges, most of the edges are already present in H.

Construction : Path Buying



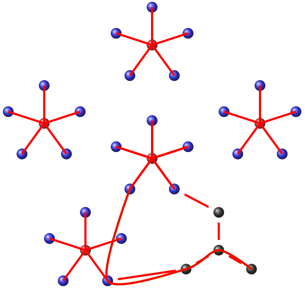
A light path has a *few* ($< n \log n / h^2$) clustered nodes

$\Downarrow$

Except a *few* ($< n \log n / h^2$) edges, most of the edges are already present in H.

- Add every light path to H.

Construction : Path Buying



A light path has a *few* ($< n \log n/h^2$) clustered nodes

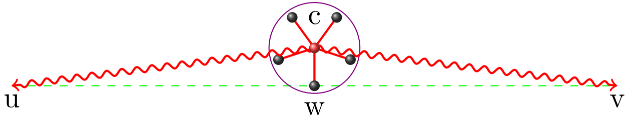
$$\Downarrow$$

Except a *few* ($< n \log n/h^2$) edges, most of the edges are already present in H.

- Add every light path to H.

Analyzing Stretch

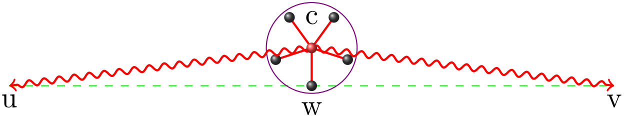
Heavy Path



Path of stretch +2 between u and v in H.

Analyzing Stretch

Heavy Path

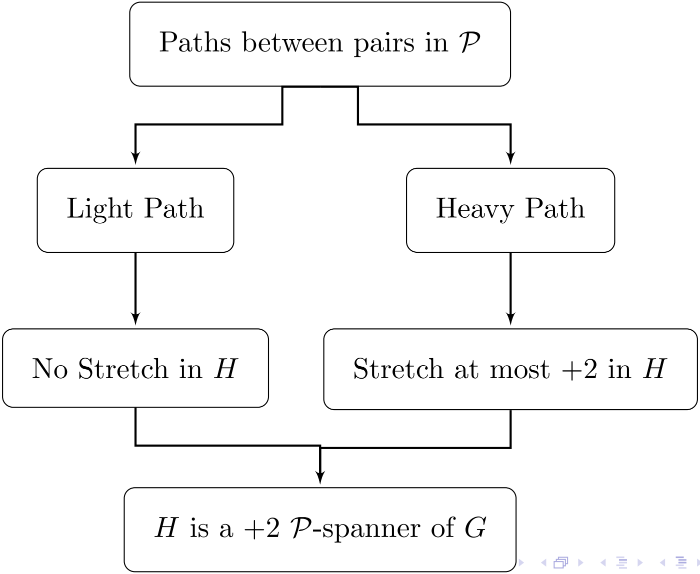


Path of stretch +2 between u and v in H.

Light Path

Path of stretch 0 between u and v in H.

Stretch of Spanner



Size of the Spanner

- **Clustering.**
  - Edges between cluster centers and clustered nodes form a forest.
  - At most nh edges incident on **unclustered** nodes, after all clusters are formed.
  - Clustering adds $O(nh)$ edges.

Size of the Spanner

- **Clustering.**
  - Edges between cluster centers and clustered nodes form a forest.
  - At most nh edges incident on **unclustered** nodes, after all clusters are formed.
  - Clustering adds $O(nh)$ edges.
- **SPT Addition.**
  - Only $O(h)$ trees are added.
  - At most $O(nh)$ edges get added.

Size of the Spanner

- **Clustering.**
  - Edges between cluster centers and clustered nodes form a forest.
  - At most nh edges incident on **unclustered** nodes, after all clusters are formed.
  - Clustering adds $O(nh)$ edges.
- **SPT Addition.**
  - Only $O(h)$ trees are added.
  - At most $O(nh)$ edges get added.
- **Adding light paths.**
  - At most $|\mathcal{P}|$ light shortest paths added.
  - Each path contributes $\leq n \log n/h^2$ edges to H.

Size of the Spanner

- **Clustering.**
  - Edges between cluster centers and clustered nodes form a forest.
  - At most nh edges incident on **unclustered** nodes, after all clusters are formed.
  - Clustering adds $O(nh)$ edges.
- **SPT Addition.**
  - Only $O(h)$ trees are added.
  - At most $O(nh)$ edges get added.
- **Adding light paths.**
  - At most $|\mathcal{P}|$ light shortest paths added.
  - Each path contributes $\leq n \log n / h^2$ edges to H.

Size of H is $O(n(|\mathcal{P}| \cdot \log n)^{1/3})$.

We have proved..

Theorem

There is a polynomial time algorithm which, given any graph $G = (V, E)$ on n nodes and any set $\mathcal{P} \subseteq V \times V$, computes a $+2$ $\mathcal{P}$-spanner of G with $\tilde{O}(n|\mathcal{P}|^{1/3})$ edges.

Conclusion

Central Question on Additive Spanners

Sparsest constant/polylogarithmic-stretch additive spanner is $+6$-spanner with $O(n^{1.33})$ edges.

Conclusion

Central Question on Additive Spanners

Sparsest constant/polylogarithmic-stretch additive spanner is $+6$-spanner with $O(n^{1.33})$ edges.

Can we do better ?

Conclusion

Central Question on Additive Spanners

Sparsest constant/polylogarithmic-stretch additive spanner is $+6$-spanner with $O(n^{1.33})$ edges.

Can we do better ?

- Our $+2$ $\mathcal{P}$-spanner is sparser for "small" enough $|\mathcal{P}|$.

Conclusion

Central Question on Additive Spanners

Sparsest constant/polylogarithmic-stretch additive spanner is $+6$-spanner with $O(n^{1.33})$ edges.

Can we do better ?

- Our $+2$ $\mathcal{P}$-spanner is sparser for "small" enough $|\mathcal{P}|$.
- Our $+2$ $S \times V$-spanner is sparser for "small" enough $|S|$.

Conclusion

Central Question on Additive Spanners

Sparsest constant/polylogarithmic-stretch additive spanner is $+6$-spanner with $O(n^{1.33})$ edges.

Can we do better ?

- Our $+2$ $\mathcal{P}$-spanner is sparser for "small" enough $|\mathcal{P}|$.
- Our $+2$ $S \times V$-spanner is sparser for "small" enough $|S|$.
- Our $+4k$ D-spanner is sparser for "large" enough D.

Thank You!!

KV13 T. Kavitha and Nithin M. Varma. Small stretch pairwise spanners. In *ICALP(1)*, pages 601-612, 2013.