CS648 Project

Jatin Jindal,
Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

CS648 Project

Jatin Jindal   Yash Mahajan

13 November, 2018

Outline

1 Deterministic Algorithm for unweighted graphs

2 Algorithm for 3-Spanner sub-graph

3 Time Complexity

4 Expected number of edges

5 Correctness

6 Algorithm for 5-Spanner sub-graph

7 Time Complexity

8 Expected number of edges

9 Correctness

10 Rough Idea for 7 spanner

Initial Ideas and Failed attempts

CS648 Project

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- For computing a sub-graph we need to remove edges from our original graph.

Initial Ideas and Failed attempts

- For computing a sub-graph we need to remove edges from our original graph.
- Now if we remove any edge there must exist another path whose length is equal to given stretch. So, the remove edges must be part of some cycle of a particular length.

Initial Ideas and Failed attempts

- For computing a sub-graph we need to remove edges from our original graph.
- Now if we remove any edge there must exist another path whose length is equal to given stretch. So, the remove edges must be part of some cycle of a particular length.
- For the case of unweighted graphs we thought of an deterministic algorithm which would try to find all edges in a cycle and try to remove them in a single BFS/DFS traversal

Initial Ideas and Failed attempts

- For computing a sub-graph we need to remove edges from our original graph.
- Now if we remove any edge there must exist another path whose length is equal to given stretch. So, the remove edges must be part of some cycle of a particular length.
- For the case of unweighted graphs we thought of an deterministic algorithm which would try to find all edges in a cycle and try to remove them in a single BFS/DFS traversal
- However that wouldn't work because we aren't able to find all cycles of a particular length, also we don't get any bound on the number of edges that remain.

Deterministic Algorithm for 3-spanner for unweighted graphs

- First we will choose some vertex as cluster centre and mark all it's neighbours as part of it's cluster and delete all intra-cluster edges.

Deterministic Algorithm for 3-spanner for unweighted graphs

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- First we will choose some vertex as cluster centre and mark all it's neighbours as part of it's cluster and delete all intra-cluster edges.
- For all of the remaining vertex we will choose just one edge from the cluster to any other vertex and delete rest all edges.

Deterministic Algorithm for 3-spanner for unweighted graphs

- First we will choose some vertex as cluster centre and mark all it's neighbours as part of it's cluster and delete all intra-cluster edges.
- For all of the remaining vertex we will choose just one edge from the cluster to any other vertex and delete rest all edges.
- We will then repeat this procedure by picking any one of the remaining unmarked vertices as cluster centres till no vertices remain unmarked.

Deterministic Algorithm for 3-spanner for unweighted graphs

- First we will choose some vertex as cluster centre and mark all it's neighbours as part of it's cluster and delete all intra-cluster edges.
- For all of the remaining vertex we will choose just one edge from the cluster to any other vertex and delete rest all edges.
- We will then repeat this procedure by picking any one of the remaining unmarked vertices as cluster centres till no vertices remain unmarked.
- This algorithm will take O(m) time.

Deterministic Algorithm for 3-spanner for unweighted graphs

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs
Algorithm for 3-Spanner sub-graph
Time Complexity
Expected number of edges
Correctness
Algorithm for 5-Spanner sub-graph
Time Complexity
Expected number of

- First we will choose some vertex as cluster centre and mark all it's neighbours as part of it's cluster and delete all intra-cluster edges.
- For all of the remaining vertex we will choose just one edge from the cluster to any other vertex and delete rest all edges.
- We will then repeat this procedure by picking any one of the remaining unmarked vertices as cluster centres till no vertices remain unmarked.
- This algorithm will take O(m) time.
- In order to do this for weighted graphs,we will take help of randomization and randomly pick the cluster centres.

Deterministic Algorithm for 3-spanner for unweighted graphs

CS648 Project

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- First we will choose some vertex as cluster centre and mark all it's neighbours as part of it's cluster and delete all intra-cluster edges.
- For all of the remaining vertex we will choose just one edge from the cluster to any other vertex and delete rest all edges.
- We will then repeat this procedure by picking any one of the remaining unmarked vertices as cluster centres till no vertices remain unmarked.
- This algorithm will take O(m) time.
- In order to do this for weighted graphs,we will take help of randomization and randomly pick the cluster centres.

Algorithm for 3-Spanner sub-graph

CS648 Project

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.

Algorithm for 3-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{\sqrt{n}}$ of picking each vertex. We will refer to them as the centers of cluster.

Algorithm for 3-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{\sqrt{n}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.

Algorithm for 3-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' $= \phi$.
- Pick a random sample of vertices with probability p $= \frac{1}{\sqrt{n}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.
- For the vertices which are not assigned to any of the clusters, add all it's edges in E'.

Algorithm for 3-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' $= \phi$.
- Pick a random sample of vertices with probability p $= \frac{1}{\sqrt{n}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.
- For the vertices which are not assigned to any of the clusters, add all it's edges in E'.
- For the assigned vertices, include all the edges in E' whose weight is smaller than the weight of the vertex to the mapped center.

Algorithm for 3-Spanner sub-graph

CS648 Project

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{\sqrt{n}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.
- For the vertices which are not assigned to any of the clusters, add all it's edges in E'.
- For the assigned vertices, include all the edges in E' whose weight is smaller than the weight of the vertex to the mapped center.
- For every vertex u, add the smallest edge in (E-E')

Time Complexity

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- Step 1 and 2 takes O(n) time.

Time Complexity

- Step 1 and 2 takes O(n) time.
- Step 3,4,5 and 6 takes O(m) time.

Time Complexity

- Step 1 and 2 takes O(n) time.
- Step 3,4,5 and 6 takes O(m) time.
- Hence, total time of the algorithm is O(n+m) = O(m) time.

Expected number of edges

- Edges in the final graph(G') consists of the cluster edges, intra-cluster edges, inter-cluster edges, and edges from the vertices which are not part of any cluster.

Expected number of edges

- Edges in the final graph(G') consists of the cluster edges, intra-cluster edges, inter-cluster edges, and edges from the vertices which are not part of any cluster.
- Now, we will find the bound in each type of edge.

Expected Number of edges
continued..

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges $= O(n)$.

Expected Number of edges
continued..

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = O(n).
- **Expected number of edges from the vertices outside the cluster:**
  Consider a vertex v and assume its degree is denoted by d. Since the vertex is not part of the cluster, therefore v and its neighbours are not picked as centers. Hence, expected contribution of edges from a vertex:

Expected Number of edges
continued..

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = O(n).
- **Expected number of edges from the vertices outside the cluster:**
  Consider a vertex v and assume its degree is denoted by d. Since the vertex is not part of the cluster, therefore v and its neighbours are not picked as centers. Hence, expected contribution of edges from a vertex:

$$\mathbf{E}[E_v] = d(1-p)^{d+1}$$

$$\frac{\partial v}{\partial d} = (1-p)^{d+1} + d(1-p)^{d+1} Ln(1-p)$$

$$\frac{\partial v}{\partial d} \geq 0 \Rightarrow d \leq \frac{-1}{Ln(1-p)} \leq \frac{1}{p} = \sqrt{n}$$

$$\Rightarrow \mathbf{E}[E_v] < \sqrt{n}$$

Expected Number of edges
continued..

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = O(n).
- **Expected number of edges from the vertices outside the cluster:**
  Consider a vertex v and assume its degree is denoted by d. Since the vertex is not part of the cluster, therefore v and its neighbours are not picked as centers. Hence, expected contribution of edges from a vertex:

$$\mathbf{E}[E_v] = d(1-p)^{d+1}$$
$$\frac{\partial v}{\partial d} = (1-p)^{d+1} + d(1-p)^{d+1} Ln(1-p)$$
$$\frac{\partial v}{\partial d} \geq 0 \Rightarrow d \leq \frac{-1}{Ln(1-p)} \leq \frac{1}{p} = \sqrt{n}$$
$$\Rightarrow \mathbf{E}[E_v] < \sqrt{n}$$

Hence, expected number of edges from vertices not part of

Expected Number of Edges
continued..

- **Inter Cluster Edges with higher weights than center:** Note that each vertex has at most one each edge to each cluster. Total number of such clusters are np thus total number of edges per vertex $= O(\sqrt{n})$ and total such edges $= O(n\sqrt{n})$.

Expected Number of Edges
continued..

- **Inter Cluster Edges with higher weights than center:** Note that each vertex has at most one each edge to each cluster. Total number of such clusters are np thus total number of edges per vertex $= O(\sqrt{n})$ and total such edges $= O(n\sqrt{n})$.
- **Edges with weight less than the edge to center:** Consider a vertex v with degree d. Hence, expected number of edges through the vertex,

Expected Number of Edges

continued..

- **Inter Cluster Edges with higher weights than center:** Note that each vertex has at most one each edge to each cluster. Total number of such clusters are np thus total number of edges per vertex $= O(\sqrt{n})$ and total such edges $= O(n\sqrt{n})$.
- **Edges with weight less than the edge to center:** Consider a vertex v with degree d. Hence, expected number of edges through the vertex,

$$\sum_{k=0}^{d} k(1-p)^k p \leq \frac{1-p}{p} \leq \frac{1}{p} = \sqrt{n}$$

Hence, expected number of edges $= O(n\sqrt{n})$.

Correctness

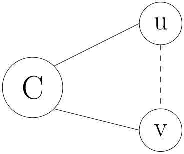
- The edges which are not present in the final graph are either the edges among the inter-cluster vertices or intra-cluster vertices.

Correctness

- The edges which are not present in the final graph are either the edges among the inter-cluster vertices or intra-cluster vertices.
- **Edge between Intracluster vertices:** Note that since the edge(u-v) is not present, therefore the distance of u and v from the centers should be less than w(u-v).

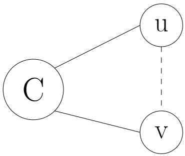
Correctness

- The edges which are not present in the final graph are either the edges among the inter-cluster vertices or intra-cluster vertices.
- **Edge between Intracluster vertices:** Note that since the edge(u-v) is not present, therefore the distance of u and v from the centers should be less than w(u-v).



Correctness

- The edges which are not present in the final graph are either the edges among the inter-cluster vertices or intra-cluster vertices.
- **Edge between Intracluster vertices:** Note that since the edge(u-v) is not present, therefore the distance of u and v from the centers should be less than w(u-v).



3-stretch holds because the weight of w(C-u) and w(C-v) is less than w(u-v) because of the way we picked the edges.

Correctness
continued..

- **Edge between intercluster vertices:**
  An inter-cluster edge is not present iff the weight of this edge is greater than their centers and also there is another edge from one of the vertex to other cluster with lower weight.

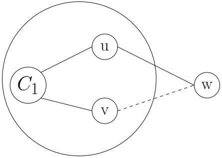
Correctness

continued..

- **Edge between intercluster vertices:**
  An inter-cluster edge is not present iff the weight of this edge is greater than their centers and also there is another edge from one of the vertex to other cluster with lower weight.
  Case 1 : $w(C_1 - u) < w(u - w) < w(v - w)$ and $w(C_1 - v) < W(v - w)$

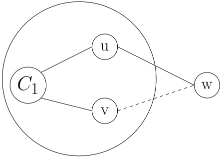
Correctness
continued..

- **Edge between intercluster vertices:**
  An inter-cluster edge is not present iff the weight of this edge is greater than their centers and also there is another edge from one of the vertex to other cluster with lower weight.
  Case 1 : $w(C_1 - u) < w(u - w) < w(v - w)$ and $w(C_1 - v) < W(v - w)$

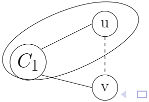


CS648 Project
Jatin Jindal, Yash Mahajan
Deterministic Algorithm for unweighted graphs
Algorithm for 3-Spanner sub-graph
Time Complexity
Expected number of edges
Correctness
Algorithm for 5-Spanner sub-graph
Time Complexity
Expected number of

- **Edge between intercluster vertices:**
  An inter-cluster edge is not present iff the weight of this edge is greater than their centers and also there is another edge from one of the vertex to other cluster with lower weight.
  Case 1 : $w(C_1 - u) < w(u - w) < w(v - w)$ and $w(C_1 - v) < W(v - w)$



  Case 2: $w(C_1 - v) < w(u - v)$ and $w(u - C_1) < w(u - v)$



Algorithm for 5-Spanner sub-graph

CS648 Project

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.

Algorithm for 5-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{n^{1/3}}$ of picking each vertex. We will refer to them as the centers of cluster.

Algorithm for 5-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{n^{1/3}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.

Algorithm for 5-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{n^{1/3}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.
- For the vertices which are not assigned to any of the clusters, add all it's edges in E'.

Algorithm for 5-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' = ϕ.
- Pick a random sample of vertices with probability p = $\frac{1}{n^{1/3}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.
- For the vertices which are not assigned to any of the clusters, add all it's edges in E'.
- For the assigned vertices, include all the edges in E' whose weight is smaller than the weight of the vertex to the mapped center.

Algorithm for 5-Spanner sub-graph

- Our algorithm will give the final graph G' = (V,E') with the desired properties. Initially E' $= \phi$.
- Pick a random sample of vertices with probability p $= \frac{1}{n^{1/3}}$ of picking each vertex. We will refer to them as the centers of cluster.
- Consider each non-picked vertex v, if it has an edges to a center then it will be assigned to that center for which weight of edge between centre and vertex is minimum. v will be called as the part of cluster to whose center it is mapped. Add all such edges to E'.
- For the vertices which are not assigned to any of the clusters, add all it's edges in E'.
- For the assigned vertices, include all the edges in E' whose weight is smaller than the weight of the vertex to the mapped center.
- Now for every pair of clusters we add the edge with the

Time Complexity

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- Step 1 and 2 takes O(n) time.

Time Complexity

- Step 1 and 2 takes O(n) time.
- Step 3,4,5 and 6 takes O(m) time.

Time Complexity

- Step 1 and 2 takes O(n) time.
- Step 3,4,5 and 6 takes O(m) time.
- Hence, total time of the algorithm is $O(n+m) = O(m)$ time.

Expected Number of edges

CS648 Project

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = $O(n)$.

Expected Number of edges

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = $O(n)$.
- **Expected number of edges from the vertices outside the cluster:**
  Consider a vertex v and assume its degree is denoted by d. Since the vertex is not part of the cluster, therefore v and its neighbours are not picked as centers. Hence, expected contribution of edges from a vertex:

Expected Number of edges

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = O(n).
- **Expected number of edges from the vertices outside the cluster:**
  Consider a vertex v and assume its degree is denoted by d. Since the vertex is not part of the cluster, therefore v and its neighbours are not picked as centers. Hence, expected contribution of edges from a vertex:

$$\mathbf{E}[E_v] = d(1-p)^{d+1}$$

$$\frac{\partial v}{\partial d} = (1-p)^{d+1} + d(1-p)^{d+1} Ln(1-p)$$

$$\frac{\partial v}{\partial d} \geq 0 \Rightarrow d \leq \frac{-1}{Ln(1-p)} \leq \frac{1}{p} = n^{1/3}$$

$$\Rightarrow \mathbf{E}[E_v] < n^{1/3}$$

Expected Number of edges

- **Cluster edges:** Each vertex is mapped to atmost one center, therfore total number of cluster edges = O(n).
- **Expected number of edges from the vertices outside the cluster:**
  Consider a vertex v and assume its degree is denoted by d. Since the vertex is not part of the cluster, therefore v and its neighbours are not picked as centers. Hence, expected contribution of edges from a vertex:

$$\mathbf{E}[E_v] = d(1-p)^{d+1}$$

$$\frac{\partial v}{\partial d} = (1-p)^{d+1} + d(1-p)^{d+1} Ln(1-p)$$

$$\frac{\partial v}{\partial d} \geq 0 \Rightarrow d \leq \frac{-1}{Ln(1-p)} \leq \frac{1}{p} = n^{1/3}$$

$$\Rightarrow \mathbf{E}[E_v] < n^{1/3}$$

Hence, expected number of edges outside the cluster are

Expected Number of Edges
continued..

- **Inter Cluster Edges with higher weights than center:** Now for each pair of cluster there will be atmost one edge. Expected number of clusters are np=O($n^{2/3}$). Total number of such edges will be $\binom{Totalclusters}{2}$ which is O($n^{4/3}$).

Expected Number of Edges

continued..

- **Inter Cluster Edges with higher weights than center:** Now for each pair of cluster there will be atmost one edge. Expected number of clusters are np=O($n^{2/3}$). Total number of such edges will be $\binom{Totalclusters}{2}$ which is O($n^{4/3}$).
- **Edges with weight less than the edge to center:** Consider a vertex v with degree d. Hence, expected number of edges through the vertex,

Expected Number of Edges
continued..

- **Inter Cluster Edges with higher weights than center:** Now for each pair of cluster there will be atmost one edge. Expected number of clusters are np=O($n^{2/3}$). Total number of such edges will be $\binom{Totalclusters}{2}$ which is O($n^{4/3}$).
- **Edges with weight less than the edge to center:** Consider a vertex v with degree d. Hence, expected number of edges through the vertex,

$$\sum_{k=0}^{d} k(1-p)^k p \leq \frac{1-p}{p} \leq \frac{1}{p} = n^{1/3}$$

Expected Number of Edges
continued..

- **Inter Cluster Edges with higher weights than center:** Now for each pair of cluster there will be atmost one edge. Expected number of clusters are np=O($n^{2/3}$). Total number of such edges will be $\binom{Totalclusters}{2}$ which is O($n^{4/3}$).
- **Edges with weight less than the edge to center:** Consider a vertex v with degree d. Hence, expected number of edges through the vertex,

$$\sum_{k=0}^{d} k(1-p)^k p \leq \frac{1-p}{p} \leq \frac{1}{p} = n^{1/3}$$

Hence, expected number of edges = $O(n * n^{1/3}) = O(n^{4/3})$.

Correctness

Deterministic Algorithm for unweighted graphs

Algorithm for 3-Spanner sub-graph

Time Complexity

Expected number of edges

Correctness

Algorithm for 5-Spanner sub-graph

Time Complexity

Expected number of

- Now in this case our algorithm for getting intra-cluster edges is similar to that of what we used for 3 spanner. So the proof of correctness will be similar in this case.

Correctness
continued..

- **Edge between inter-cluster vertices:**
  Now in this case two 3 clusters will be connected by a singe edge. Now there exists a path with the number of edges atmost 5 between the two vertices. Two each for both the clusters and one edge connecting the clusters.

Correctness
continued..

- **Edge between inter-cluster vertices:**
  Now in this case two 3 clusters will be connected by a singe edge. Now there exists a path with the number of edges atmost 5 between the two vertices. Two each for both the clusters and one edge connecting the clusters. Now according to the constraints for (u,v), the weight of the edge connecting the clusters $<$ w(u,v).

Correctness
continued..

- **Edge between inter-cluster vertices:**
  Now in this case two 3 clusters will be connected by a singe edge. Now there exists a path with the number of edges atmost 5 between the two vertices. Two each for both the clusters and one edge connecting the clusters. Now according to the constraints for (u,v), the weight of the edge connecting the clusters $<$ w(u,v).
- Also, if the path is (u,C1,x1,x2,C2,v) where u and x1 are in cluster-1 and v and x2 are in cluster-2; w(u,C1) and w(v,C2) $<$ w(u,v). Also,w(C1,x1) and w(C2,x2) $<$w(x1,x2) $<$w(u,v). Thus all edges on the path are less than w(u,v) and stretch property is maintained for 5 spanner.

Rough Idea for 7 spanner

- First we will repeat all steps done for 3-spanner except the step of creating inter-cluster edges, with probability p= $\frac{1}{n^{1/4}}$

Rough Idea for 7 spanner

Jatin Jindal, Yash Mahajan

Deterministic Algorithm for unweighted graphs
Algorithm for 3-Spanner sub-graph
Time Complexity
Expected number of edges
Correctness
Algorithm for 5-Spanner sub-graph
Time Complexity
Expected number of

- First we will repeat all steps done for 3-spanner except the step of creating inter-cluster edges, with probability p= $\frac{1}{n^{1/4}}$
- Now we will repeat step-1 but instead of picking vertices, we will pick clusters created by previous step and treat them as a single vertex now.

Rough Idea for 7 spanner

- First we will repeat all steps done for 3-spanner except the step of creating inter-cluster edges, with probability p$=\frac{1}{n^{1/4}}$
- Now we will repeat step-1 but instead of picking vertices, we will pick clusters created by previous step and treat them as a single vertex now.
- Now the newly formed clusters will have depth atmost 2. Now we will include the minimum edge between the clusters from step-1 and step-2.

Rough Idea for 7 spanner

- First we will repeat all steps done for 3-spanner except the step of creating inter-cluster edges, with probability p$=\frac{1}{n^{1/4}}$
- Now we will repeat step-1 but instead of picking vertices, we will pick clusters created by previous step and treat them as a single vertex now.
- Now the newly formed clusters will have depth atmost 2. Now we will include the minimum edge between the clusters from step-1 and step-2.
- We will see that the clusters of step-1 and 2 have depths 1 and 2 respectively. And for every pair of vertices there will exist a path with the number of edges atmost 7(2+4+1(edge joining the clusters))