

Sparse geometric graphs with small dilation

Boris Aronov

Polytechnic University New York

Mark de Berg

Eindhoven University of Technology

Otfried Cheong

Korea Advanced Institute of Science and Technology

Joachim Gudmundsson

National ICT Australia

Herman Haverkort

Eindhoven University of Technology

Michiel Smid

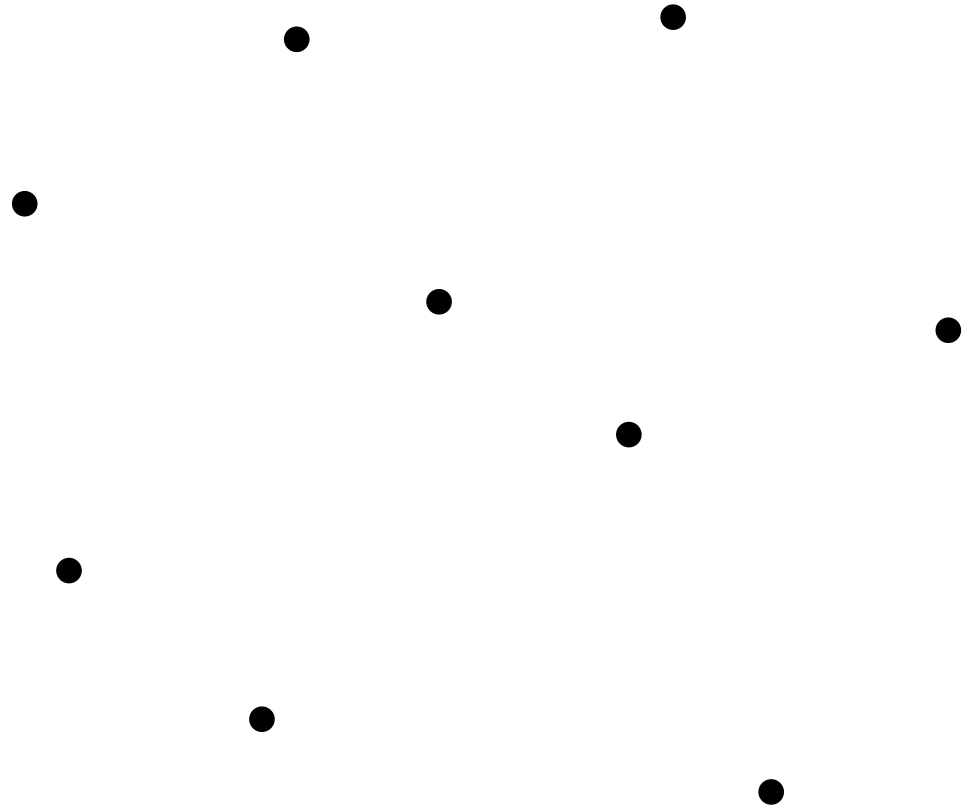
Carleton University, Ottawa

Antoine Vigneron

National University of Singapore

Spanners

Let S be a set of n points in the plane



Spanners

Let S be a set of n points in the plane

Let G be a graph with
vertex set S and straight edges

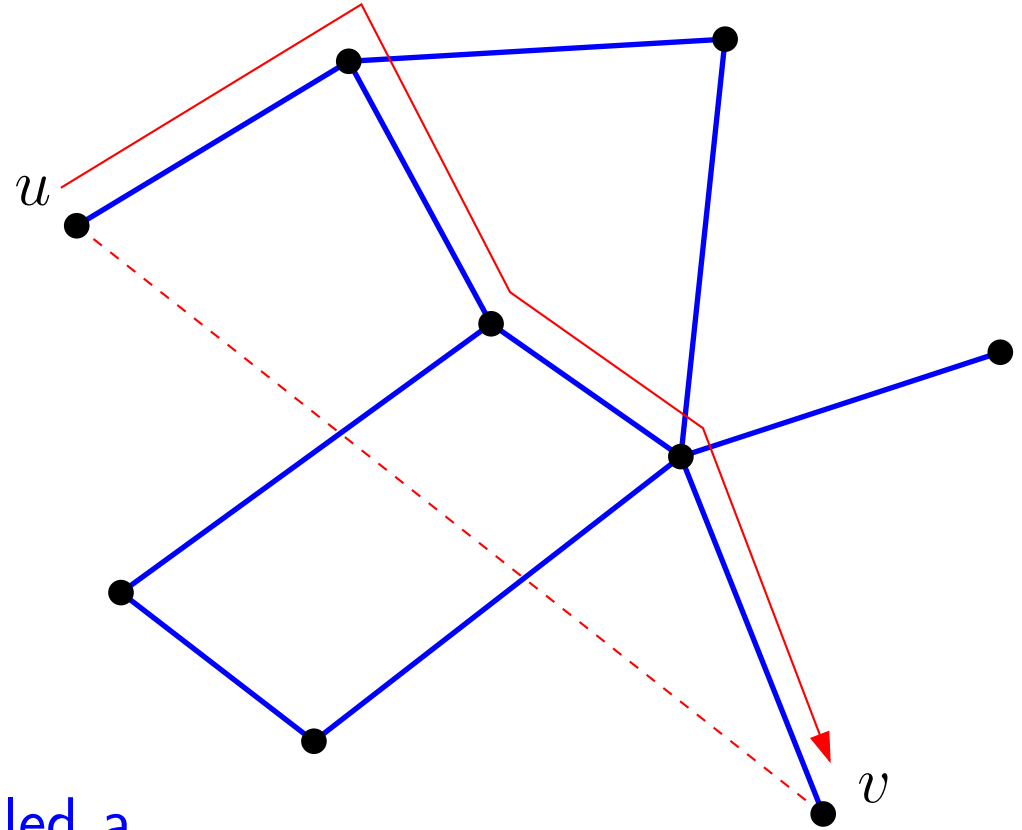
For two points $u, v \in S$, let

$$\text{dil}_G(u, v) := \frac{d_G(u, v)}{\|uv\|}$$

The dilation of G is:

$$\text{dil}(G) := \max_{u, v \in S} \text{dil}_G(u, v)$$

A graph G with $\text{dil}(G) \leq t$ is called a
 t -spanner.



Spanners

Known results on t -spanners (graphs G with $\text{dil}(G) \leq t$):

- Delaunay triangulation on S has $\leq 3n - 6$ edges, and dilation at most 2.42.
- For every set of n points S and every $\varepsilon > 0$ we can construct a $(1 + \varepsilon)$ -spanner for S with $O(n)$ edges.

Our scope: spanners with very few edges, that is:

only $n - 1 + k$ edges, for some parameter k with $0 \leq k < 2n$.

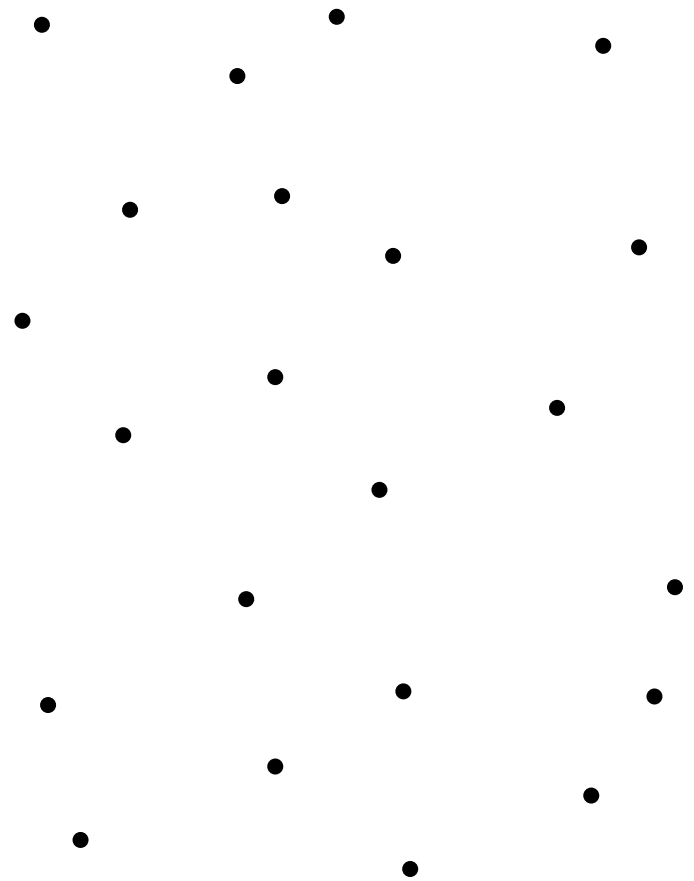
Main result: Given k and a set of n points S , ...

- our $O(n \log n)$ -time algorithm finds a spanner on S with at most $n - 1 + k$ edges and dilation $O(\frac{n}{k+1})$
- In the worst case, any graph with $n - 1 + k$ edges on S has dilation at least $\frac{2n}{\pi(k+1)}$.

$n =$ nr of points; $k =$ nr of added edges;

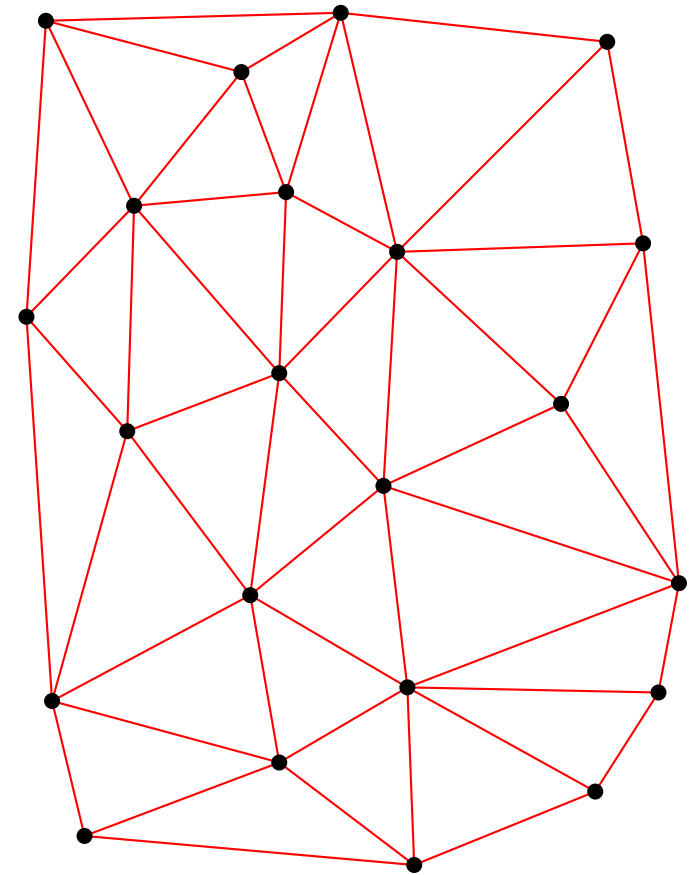
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



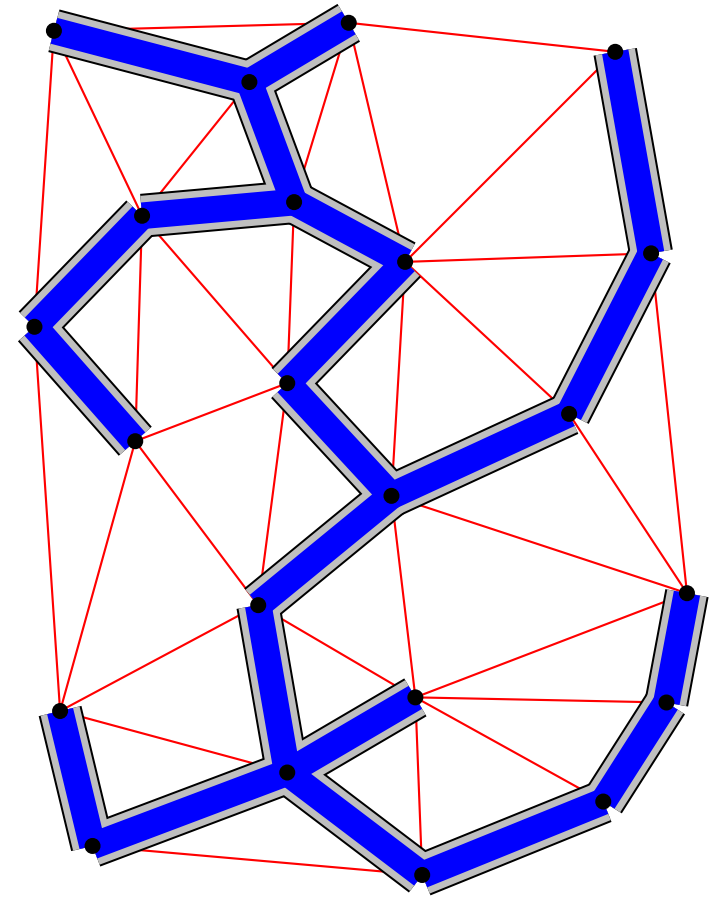
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

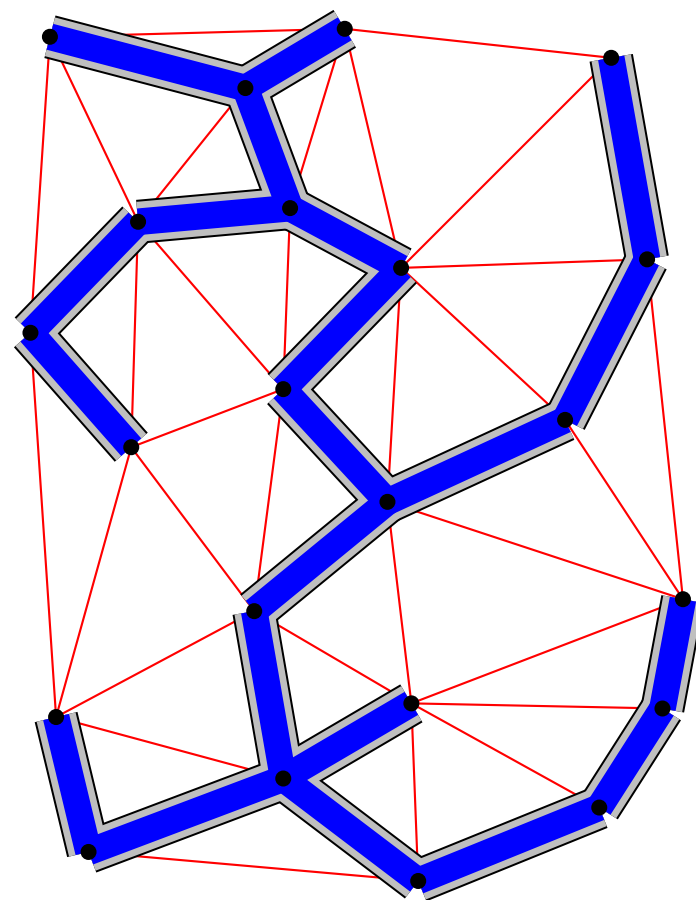
1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



n = nr of points; k = nr of added edges;

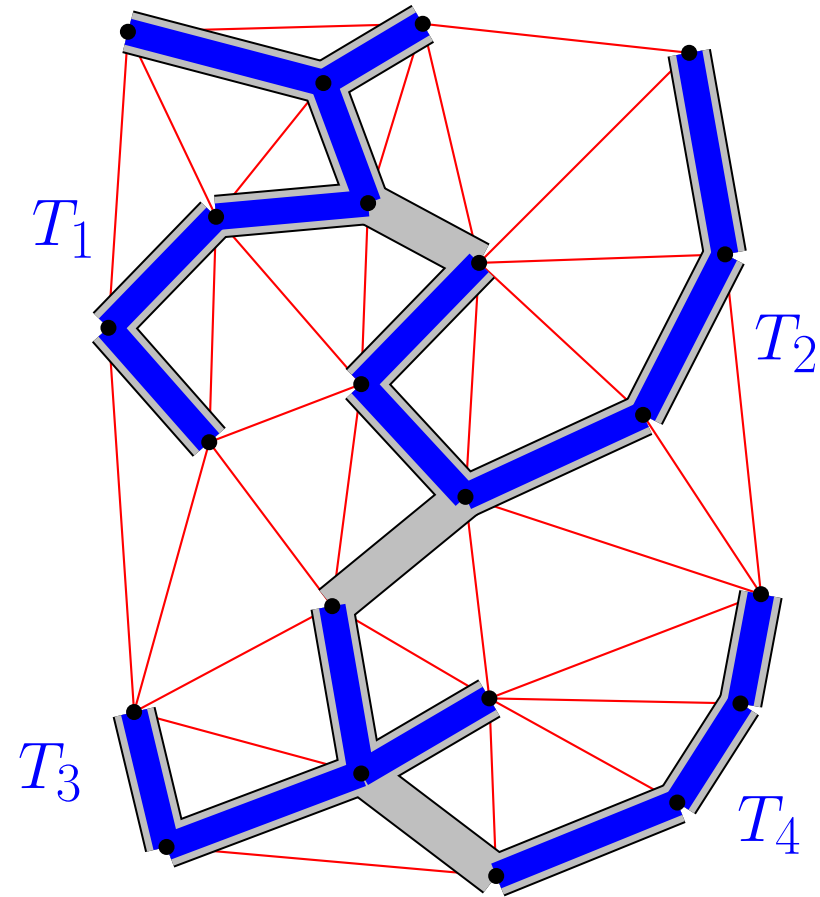
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



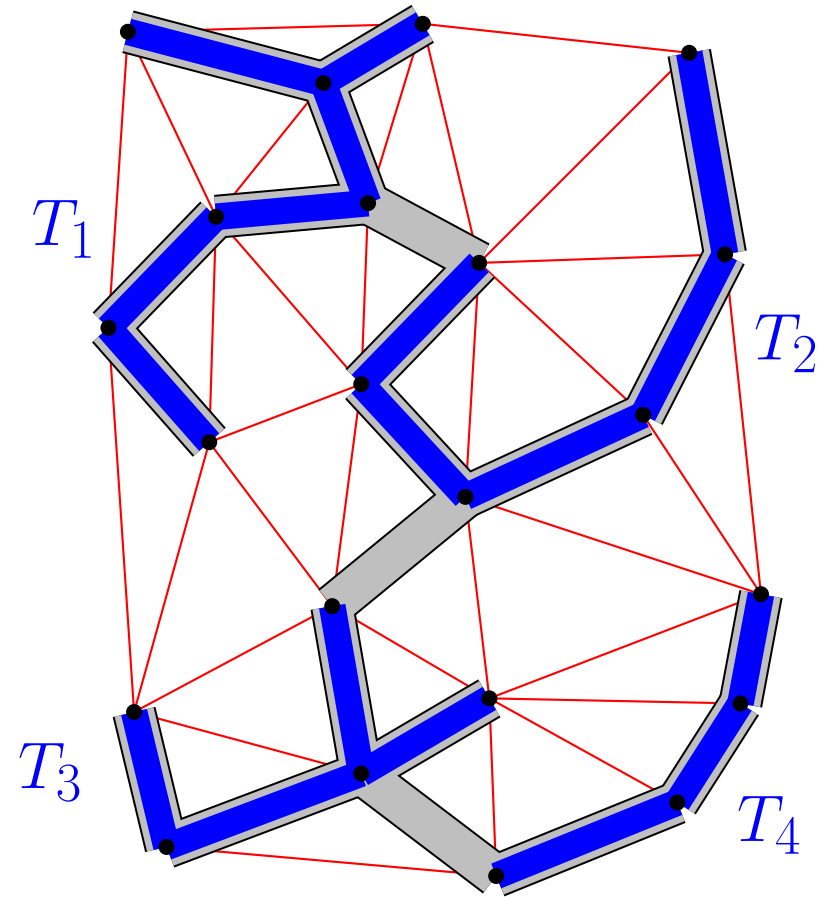
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



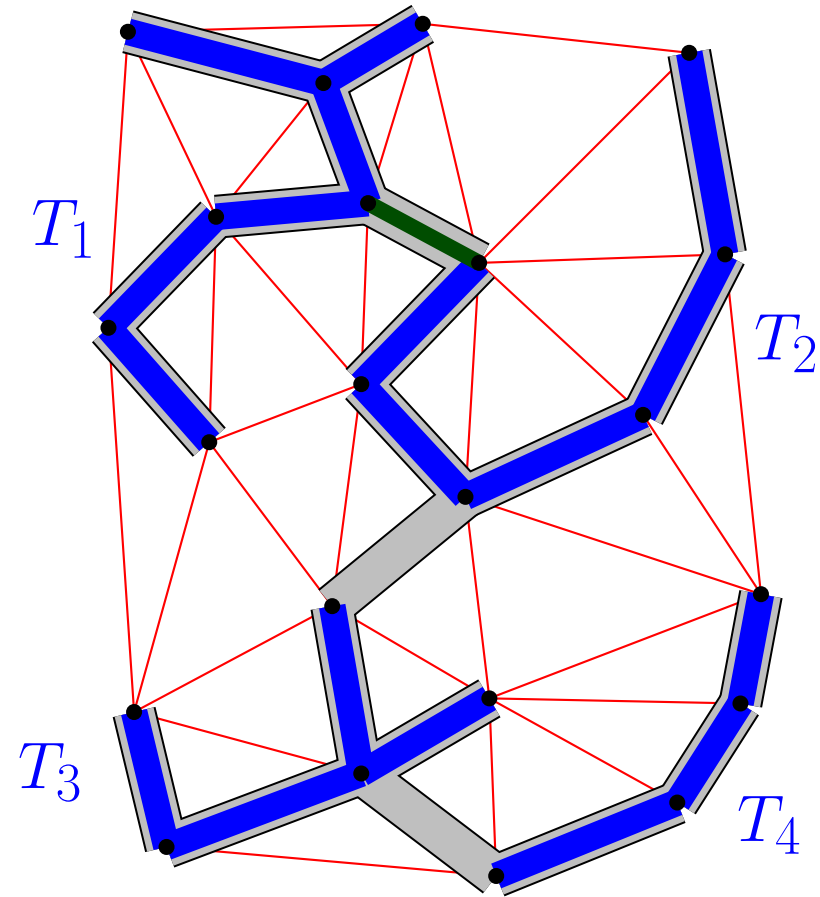
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



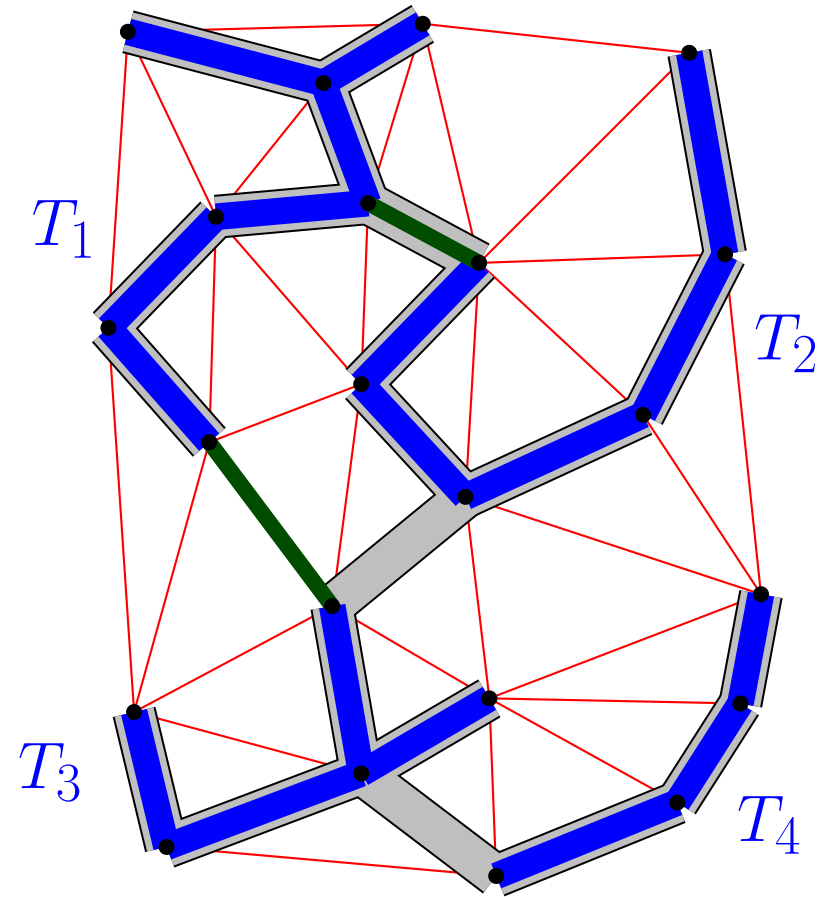
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



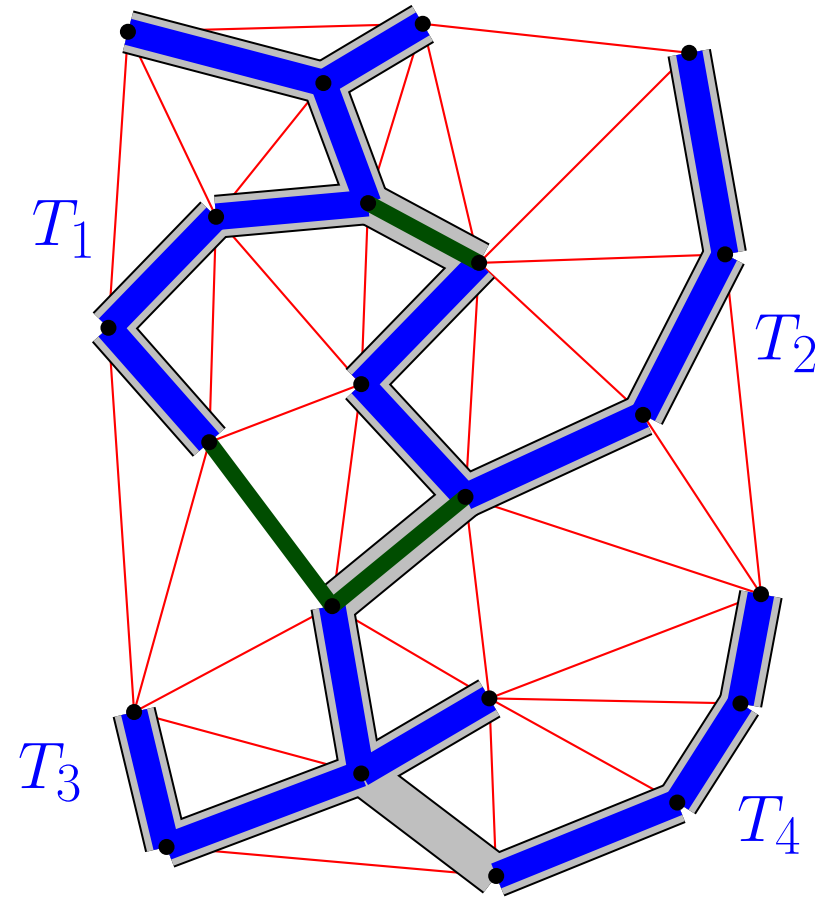
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



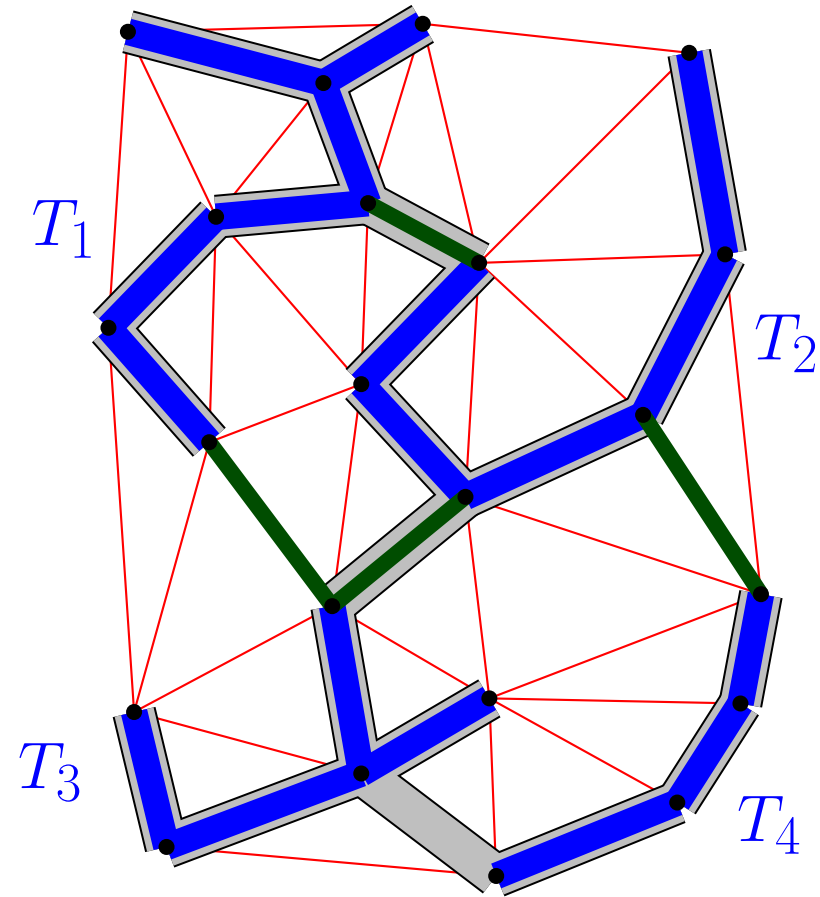
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



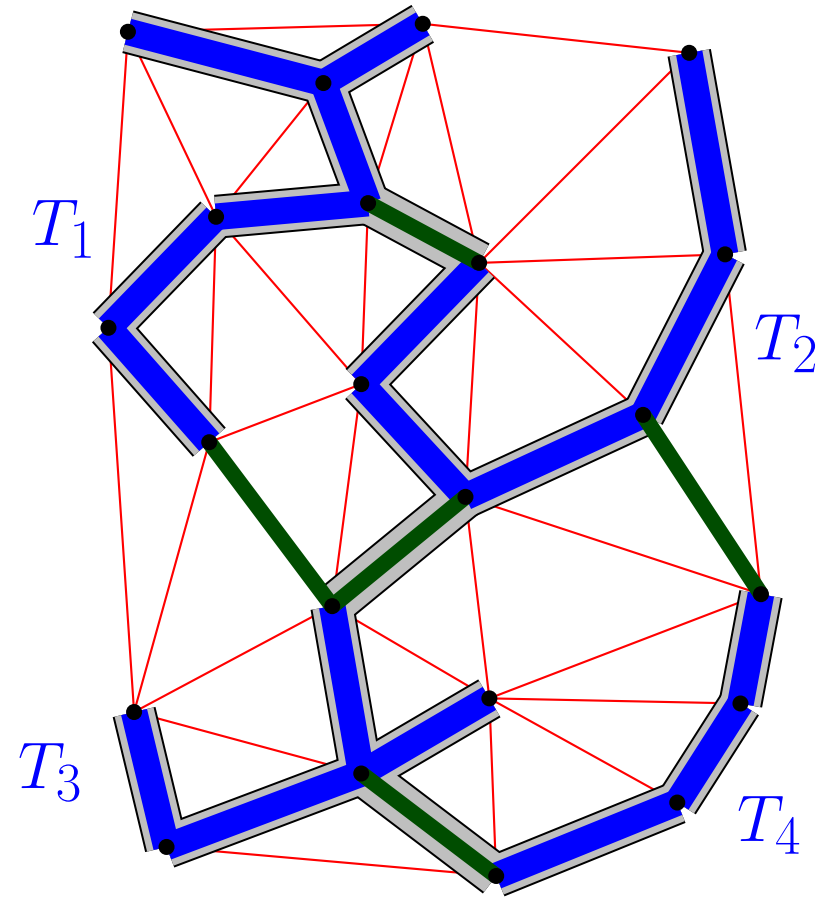
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



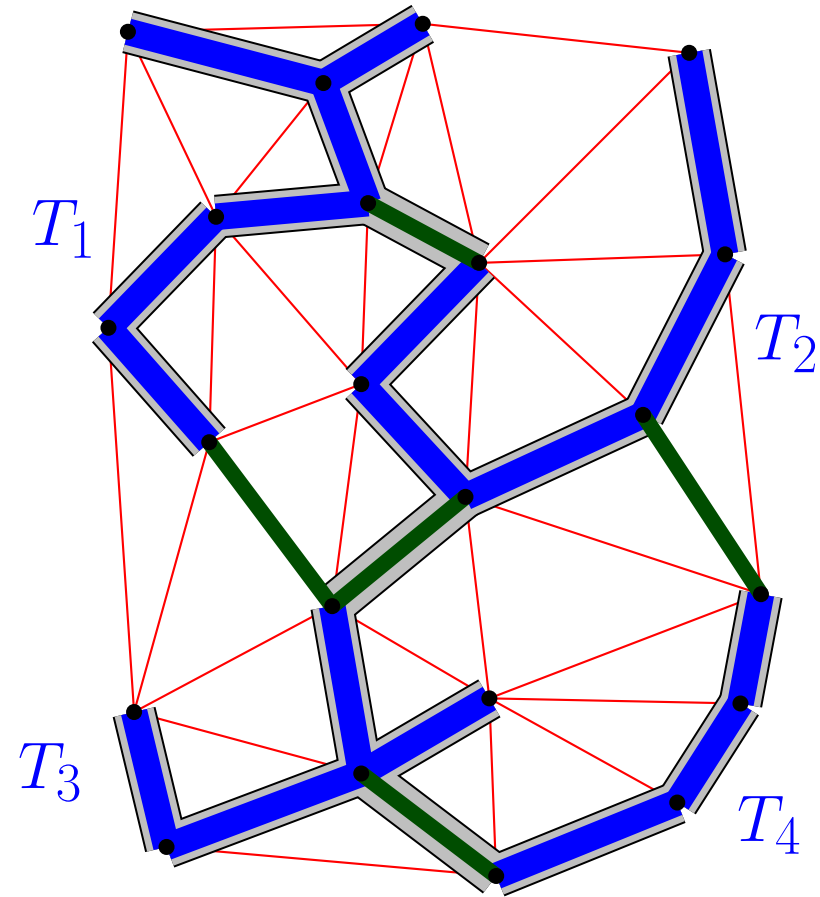
Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces:
add shortest Delaunay edge between T_i and T_j to T (if it exists)



Running time: $O(\log n)$

Nr of edges $\leq (n - 1) - (m - 1) + (3m - 6) = (n - 1) + (2m - 5) \leq n - 1 + k$

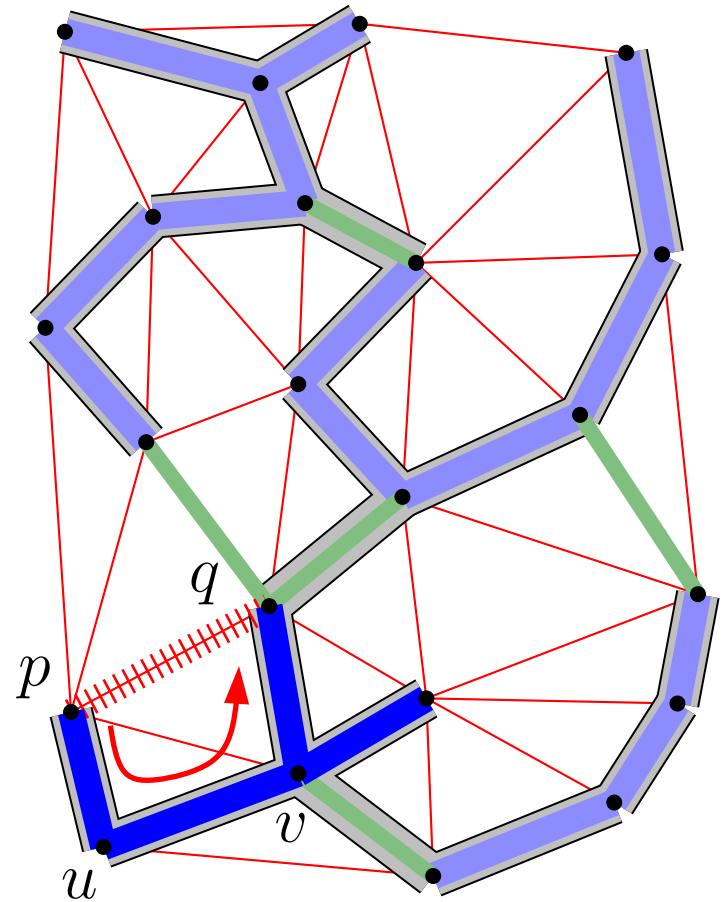
n = nr of points; k = nr of added edges;

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;



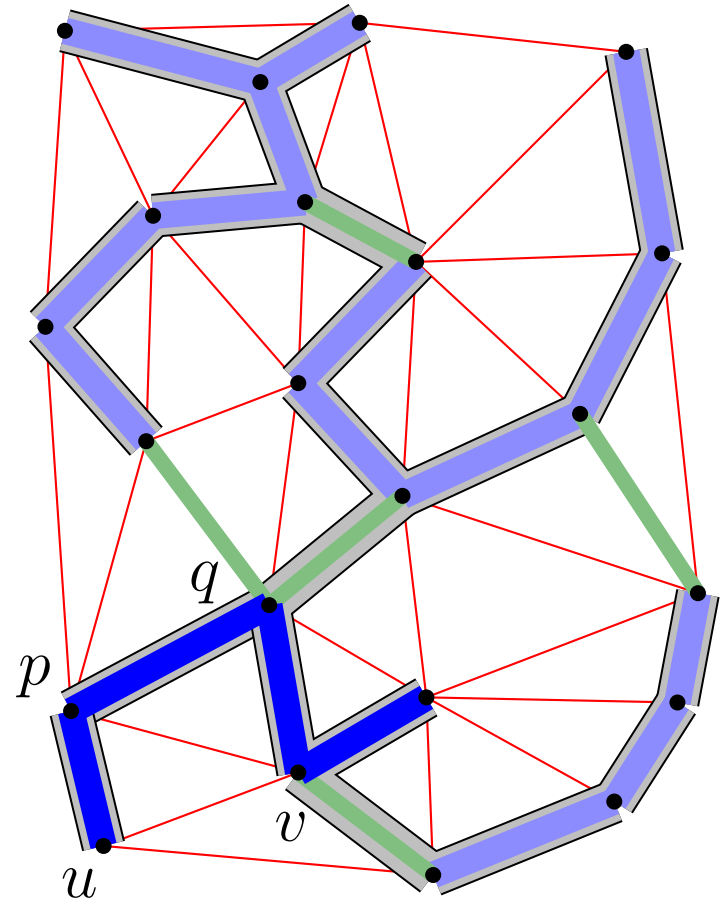
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;



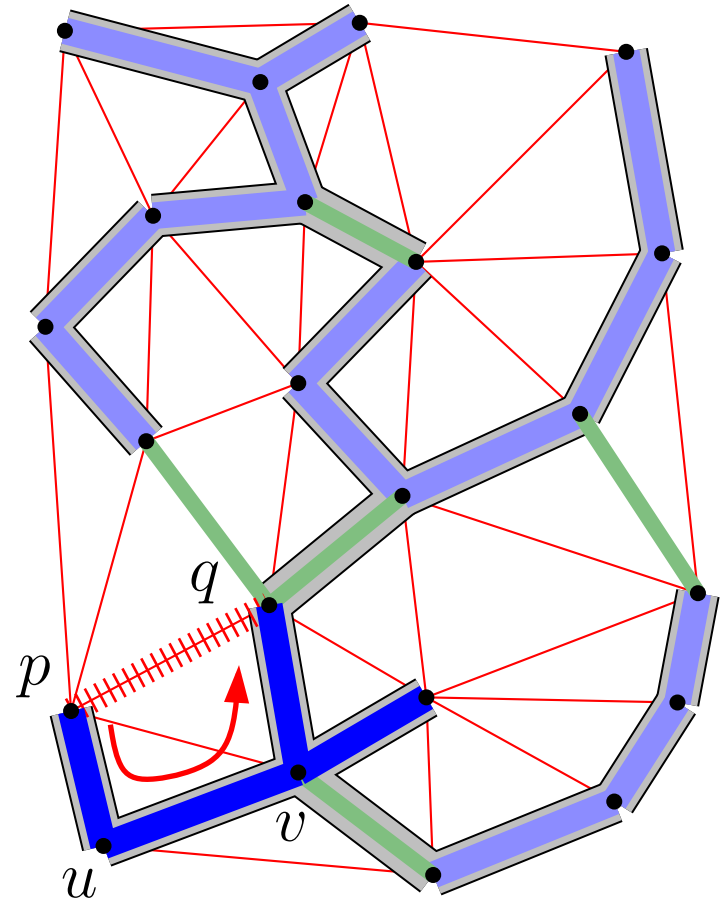
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;



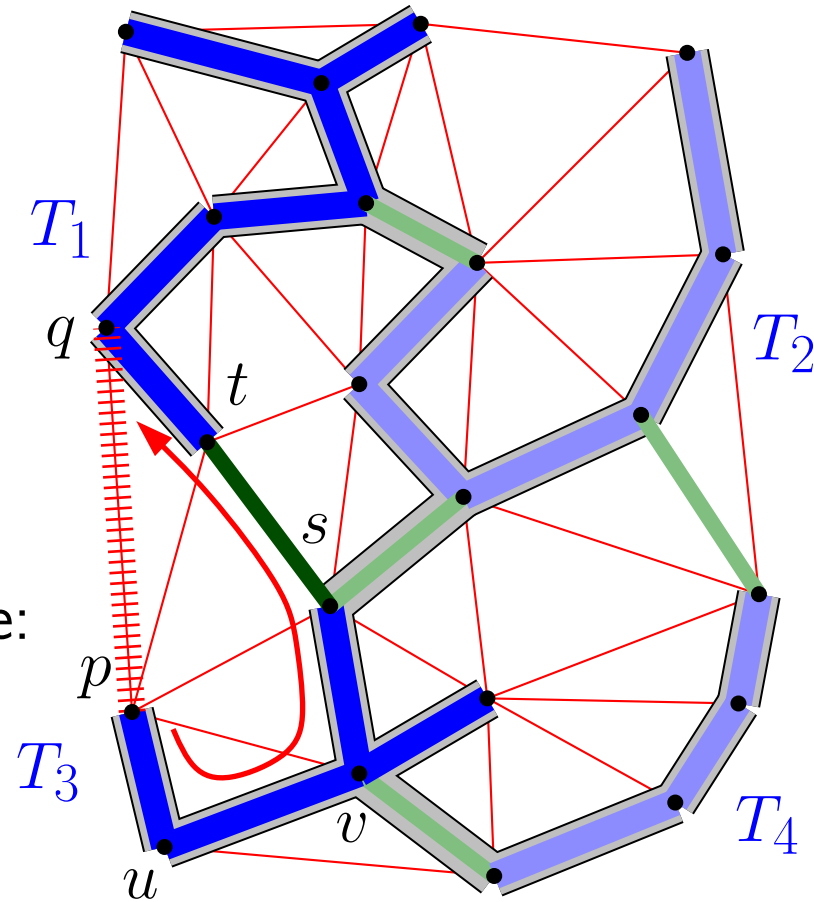
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;
- if in different pieces T_i, T_j , every edge on path in T_i , edge to T_j , path in T_j has length at most $\|pq\|$;



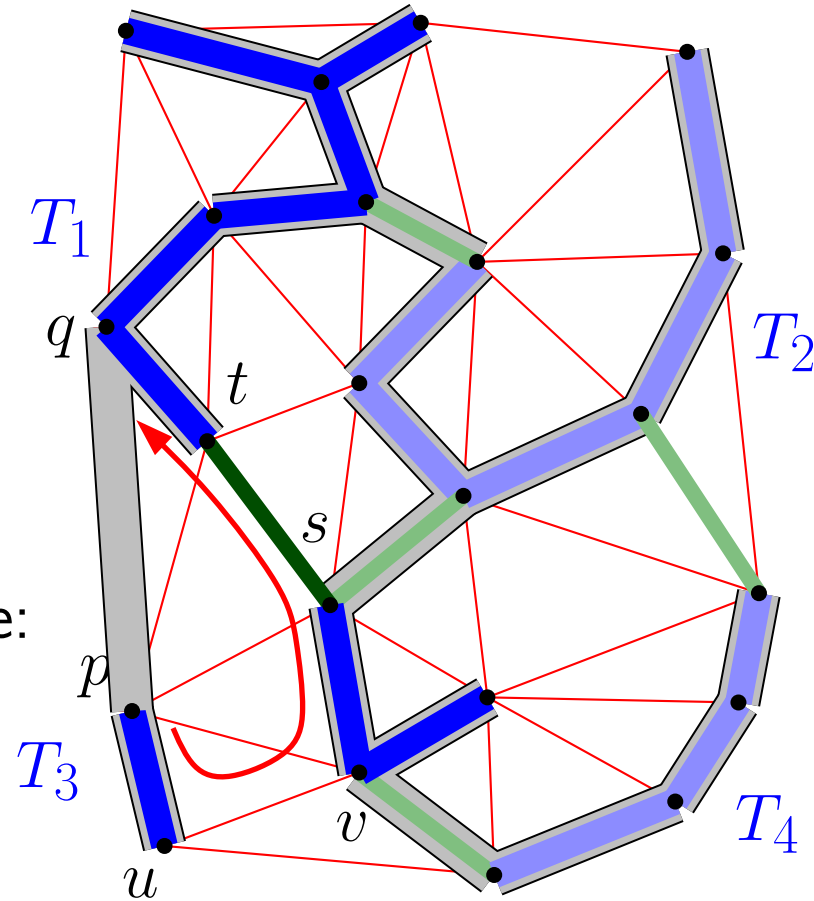
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;
- if in different pieces T_i, T_j , every edge on path in T_i , edge to T_j , path in T_j has length at most $\|pq\|$;



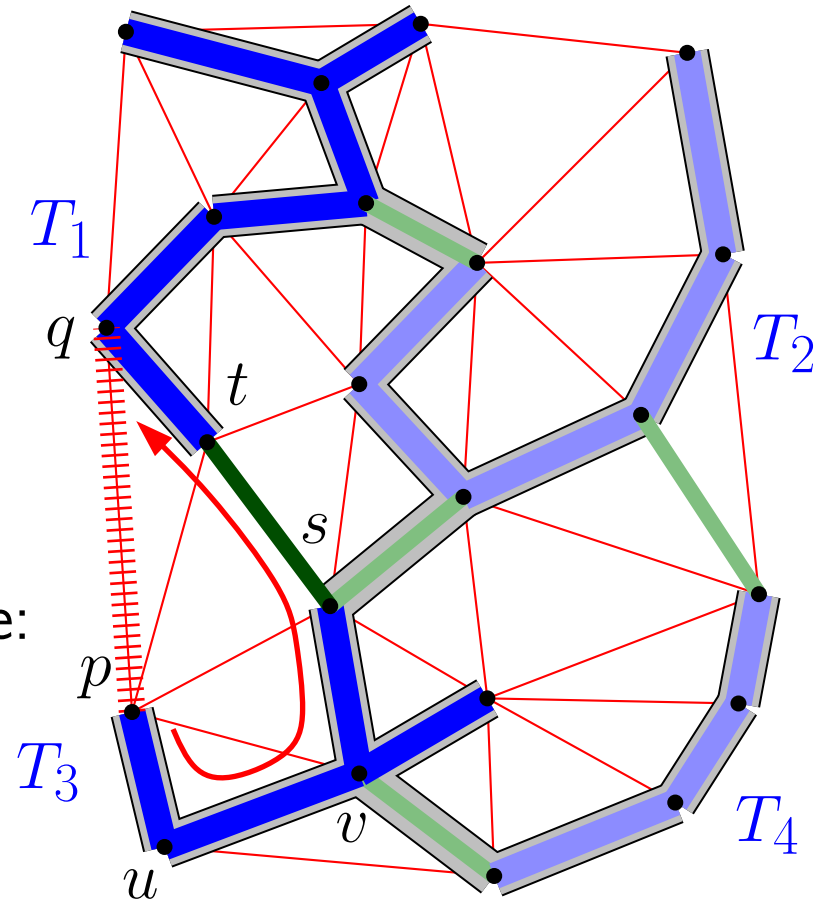
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;
- if in different pieces T_i, T_j , every edge on path in T_i , edge to T_j , path in T_j has length at most $\|pq\|$;



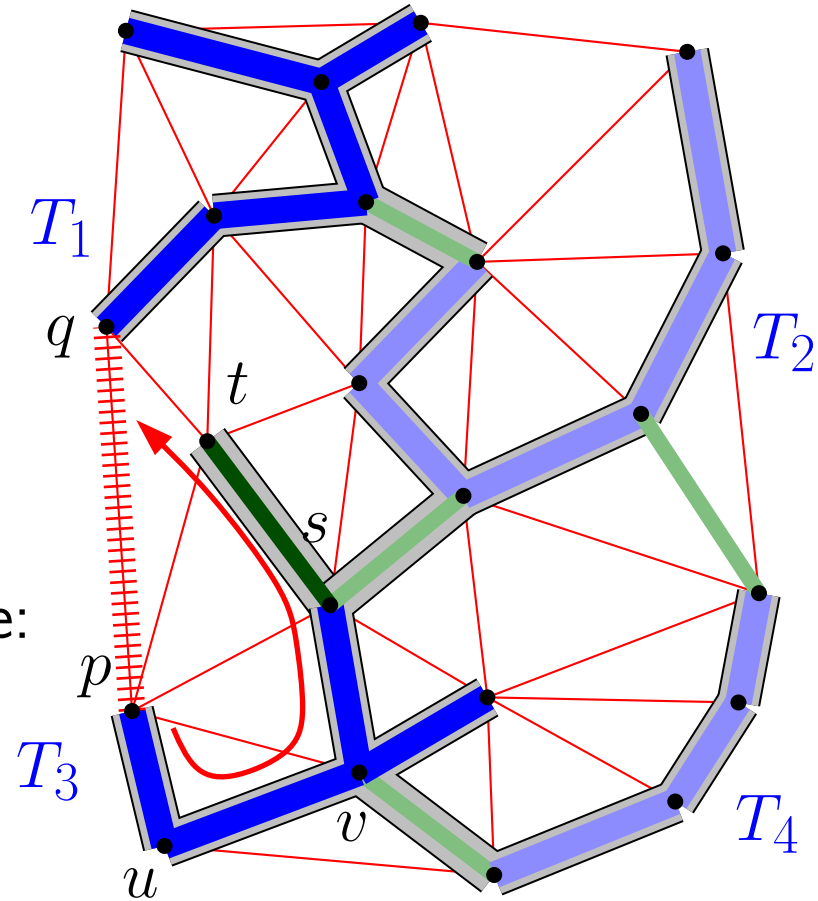
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;
- if in different pieces T_i, T_j , every edge on path in T_i , edge to T_j , path in T_j has length at most $\|pq\|$;



$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Algorithm: worst-case optimal spanner with $n - 1 + k$ edges

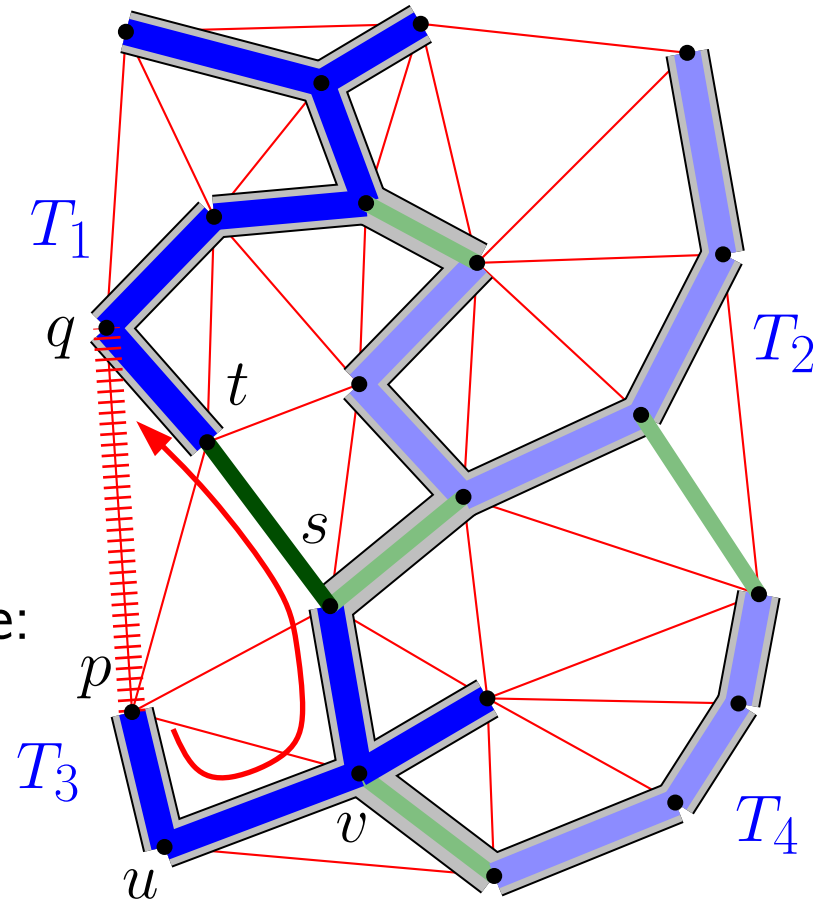
1. Compute Delaunay triangulation
2. Compute minimum spanning tree T
3. $m \leftarrow \lfloor \frac{k+5}{2} \rfloor$ (e.g. $k = 3, m = 4$)
4. Remove $m - 1$ edges from T to obtain m pieces T_1, \dots, T_m with $O(n/m)$ edges
5. For each pair T_i, T_j of pieces: add shortest Delaunay edge between T_i and T_j to T (if it exists)

Dilation for end points p, q of a Delaunay edge:

- if p, q in same piece T_i , every edge on path in T_i has length at most $\|pq\|$;
- if in different pieces T_i, T_j , every edge on path in T_i , edge to T_j , path in T_j has length at most $\|pq\|$; \rightarrow dilation $O(n/m)$.

$$d_T(p, q) \leq O(n/m)d_{DelT}(p, q) \leq O(n/m)\|pq\| = O\left(\frac{n}{k+1}\right)\|pq\|$$

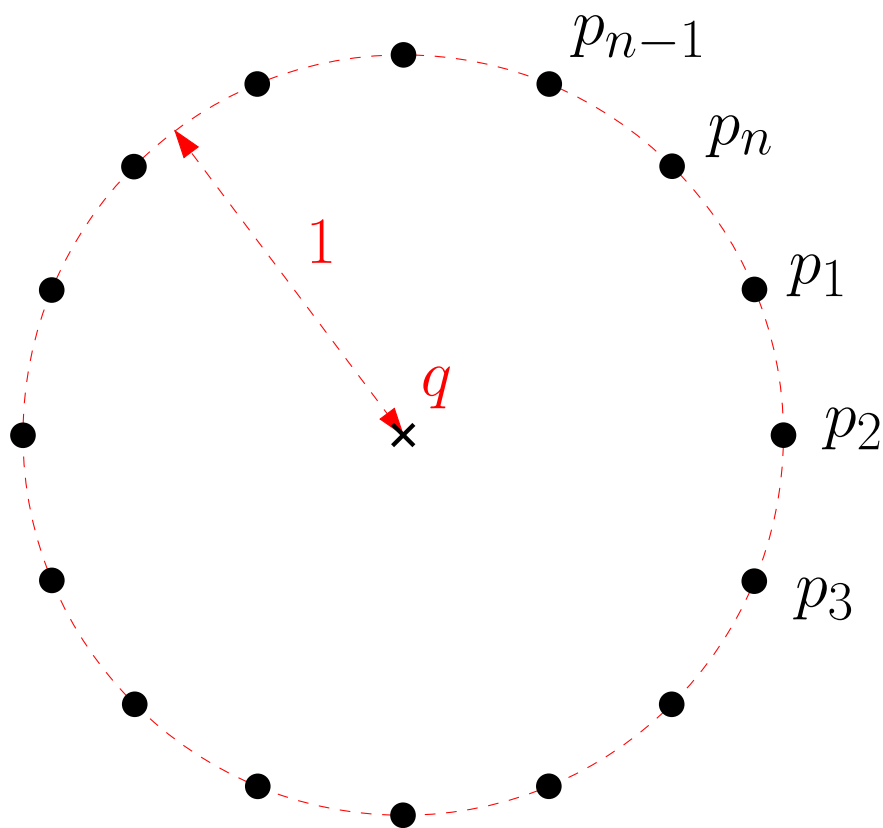
$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q)/\|p, q\|$$



Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

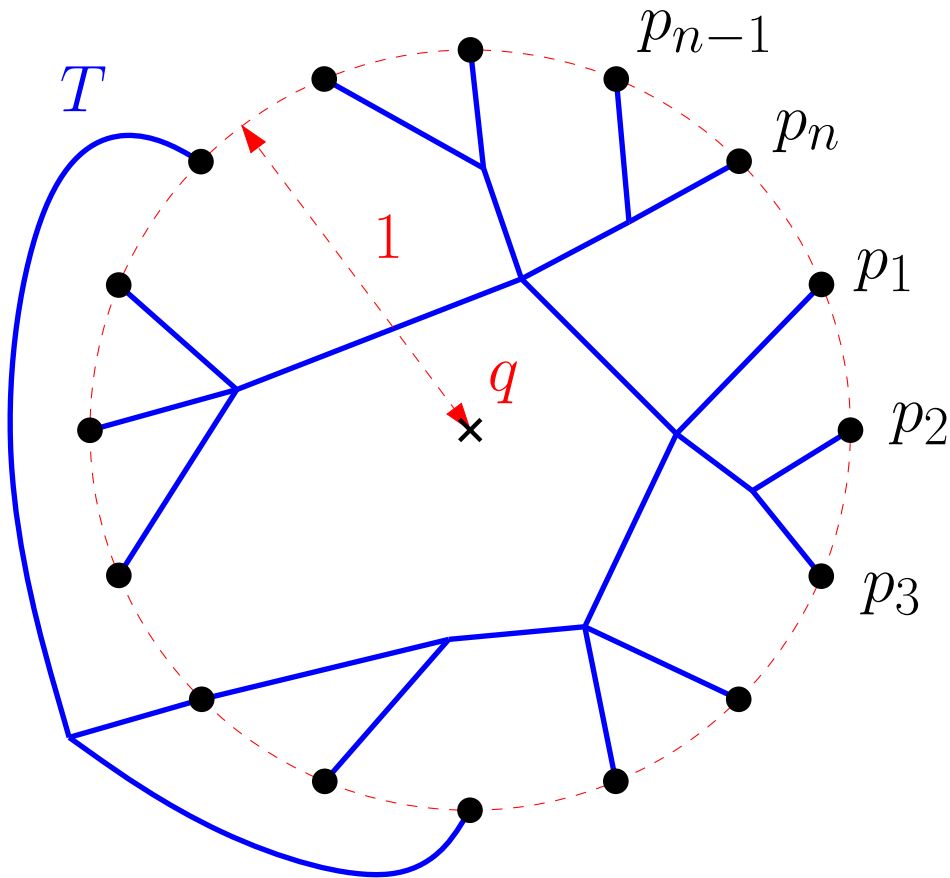


$n, k = |\text{points}|, |\text{added edges}|$; $d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T$; $\text{dil}_T(p, q) = d_T(p, q)/\|p, q\|$

Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.



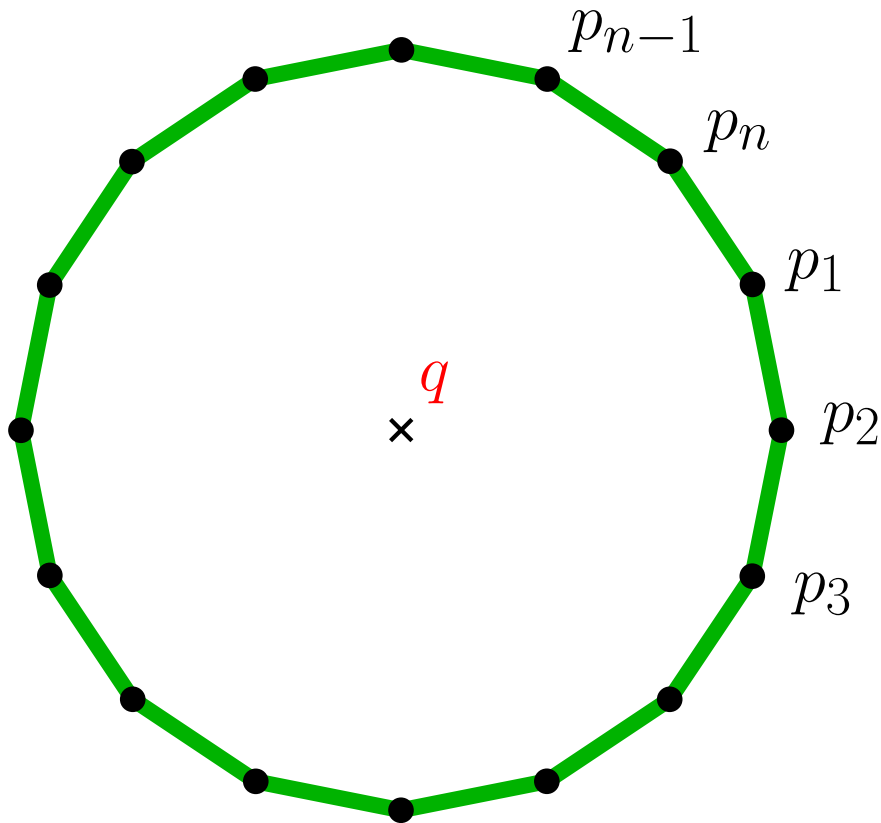
$n, k = |\text{points}|, |\text{added edges}|$; $d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T$; $\text{dil}_T(p, q) = d_T(p, q)/\|p, q\|$

Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $R^2 \setminus \{q\}$.



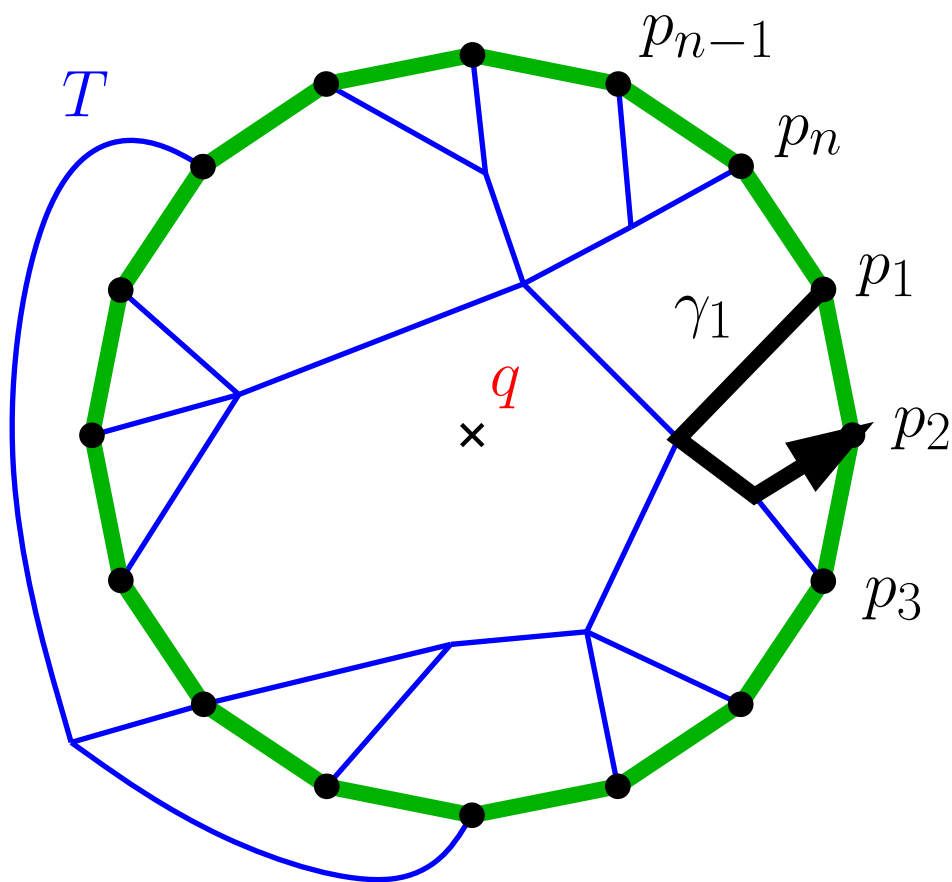
Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $R^2 \setminus \{q\}$.

Let γ_i be path from p_i to p_{i+1} in T .



$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

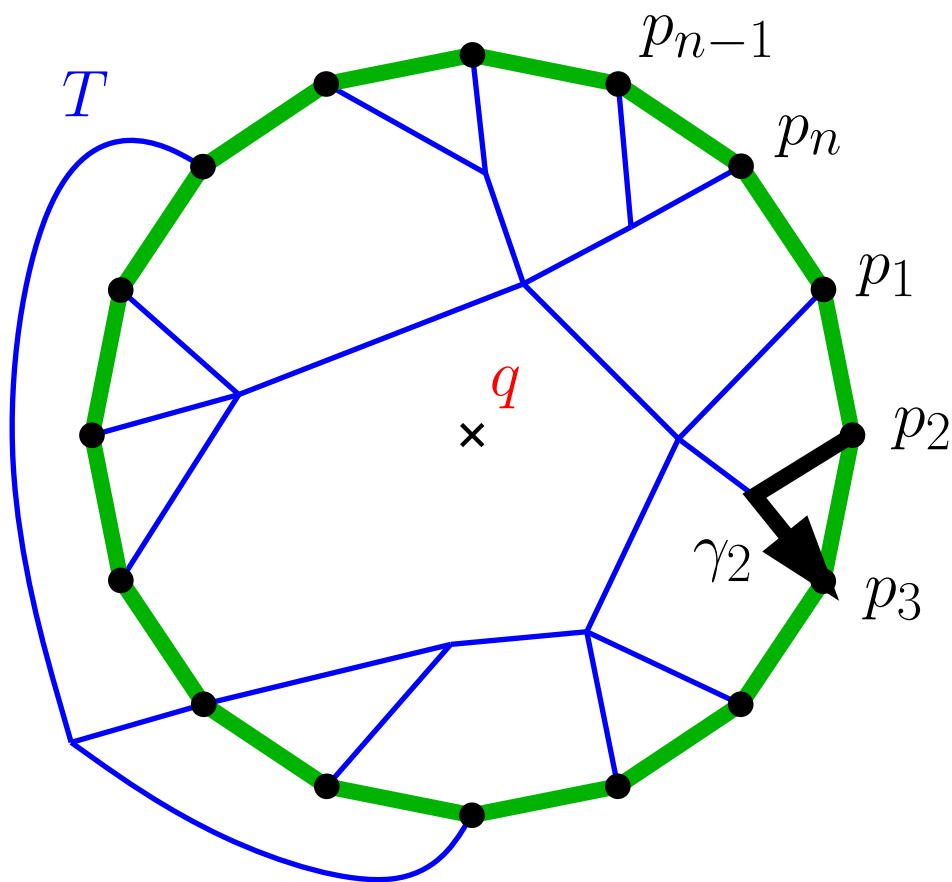
Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $\mathbb{R}^2 \setminus \{q\}$.

Let γ_i be path from p_i to p_{i+1} in T .



$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

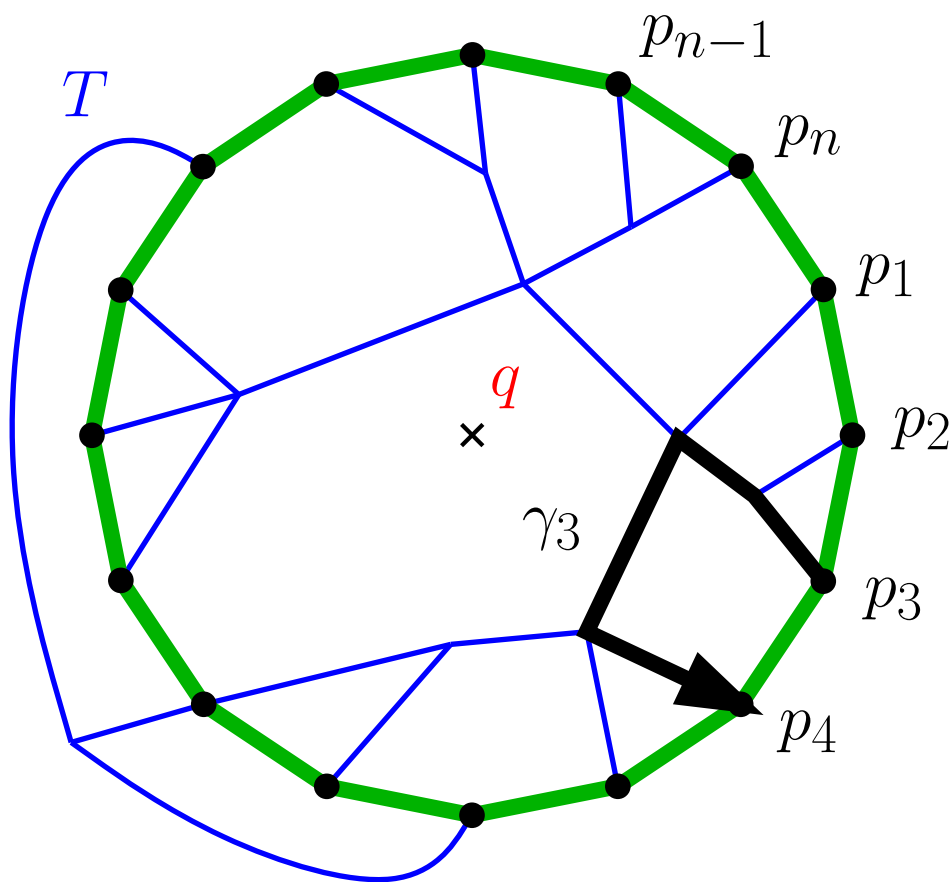
Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

Proof: The loop $p_1 p_2 p_3 \dots p_n p_1$ is not contractible in $\mathbb{R}^2 \setminus \{q\}$.

Let γ_i be path from p_i to p_{i+1} in T .



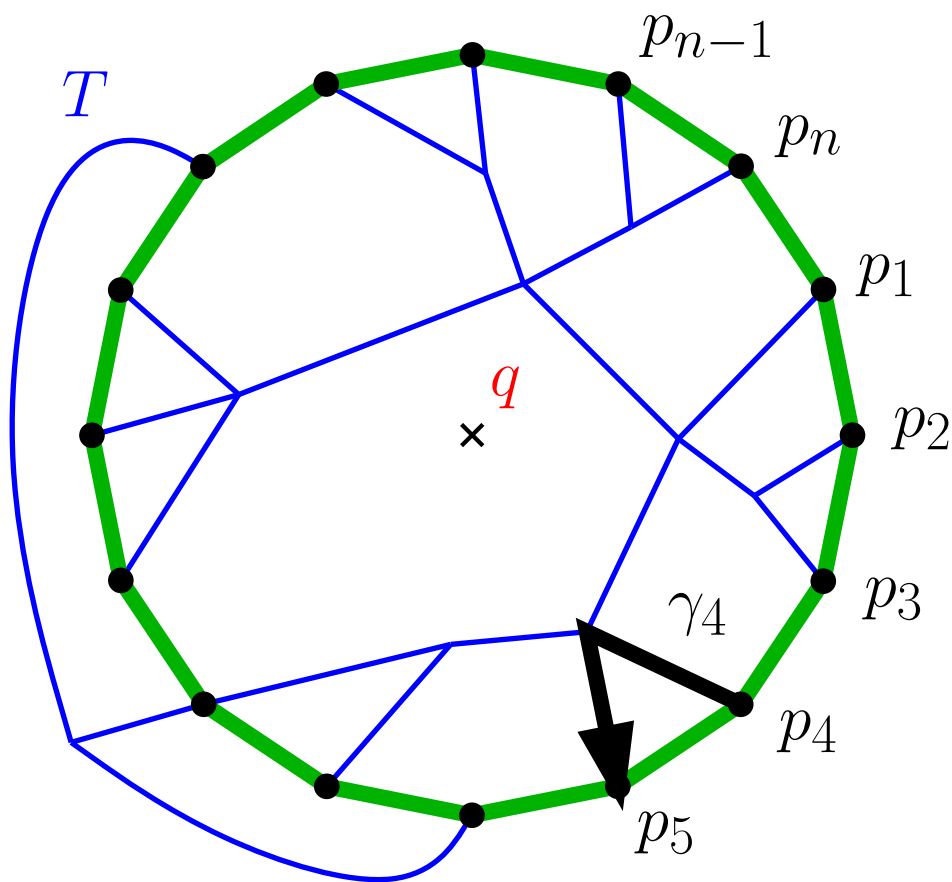
Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $\mathbb{R}^2 \setminus \{q\}$.

Let γ_i be path from p_i to p_{i+1} in T .



$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

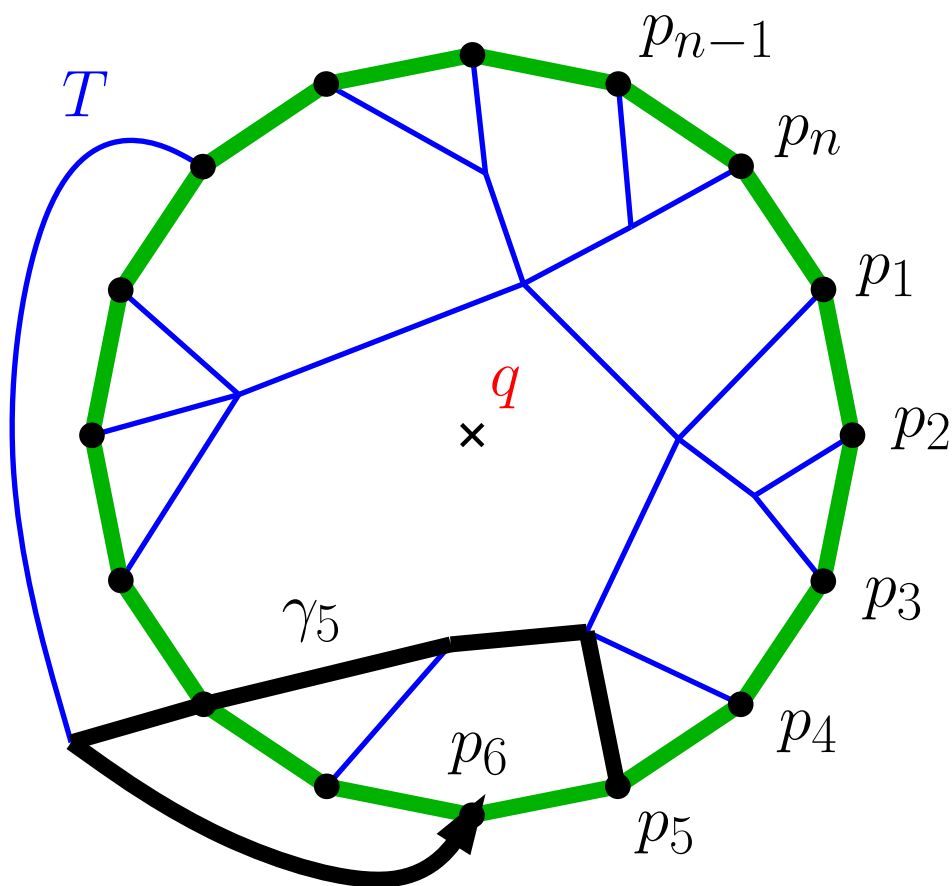
Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $\mathbb{R}^2 \setminus \{q\}$.

Let γ_i be path from p_i to p_{i+1} in T .

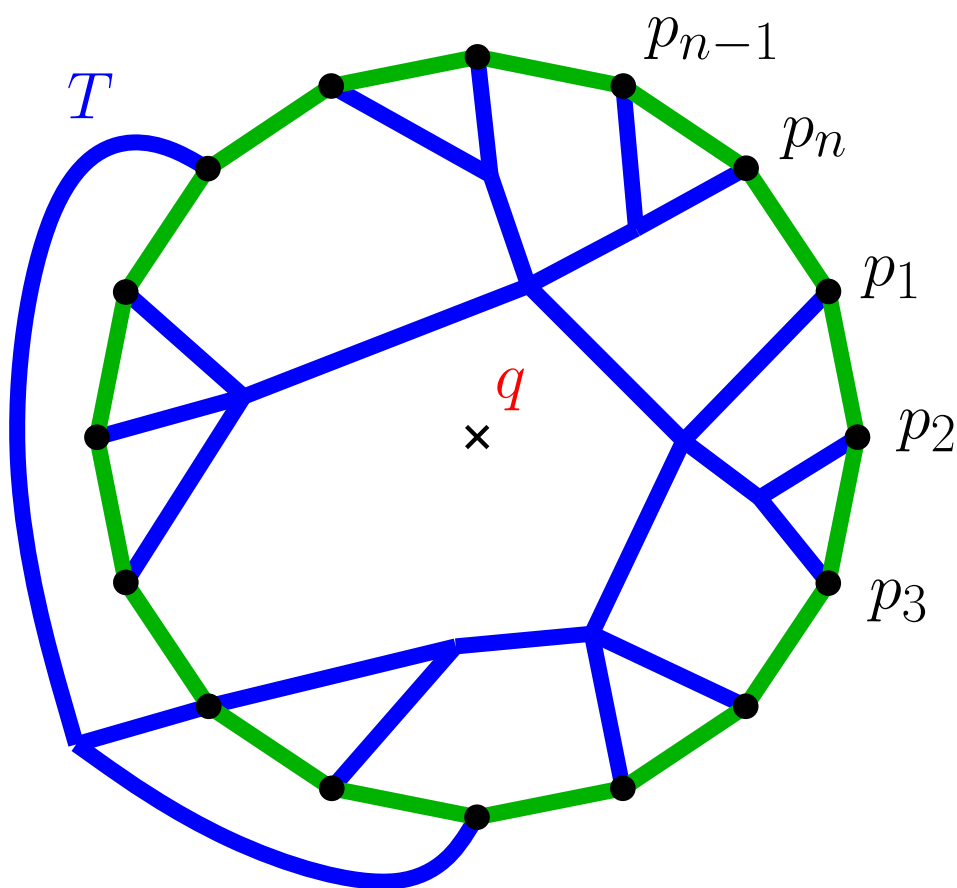


$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q) / \|p, q\|$$

Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.



Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $R^2 \setminus \{q\}$.

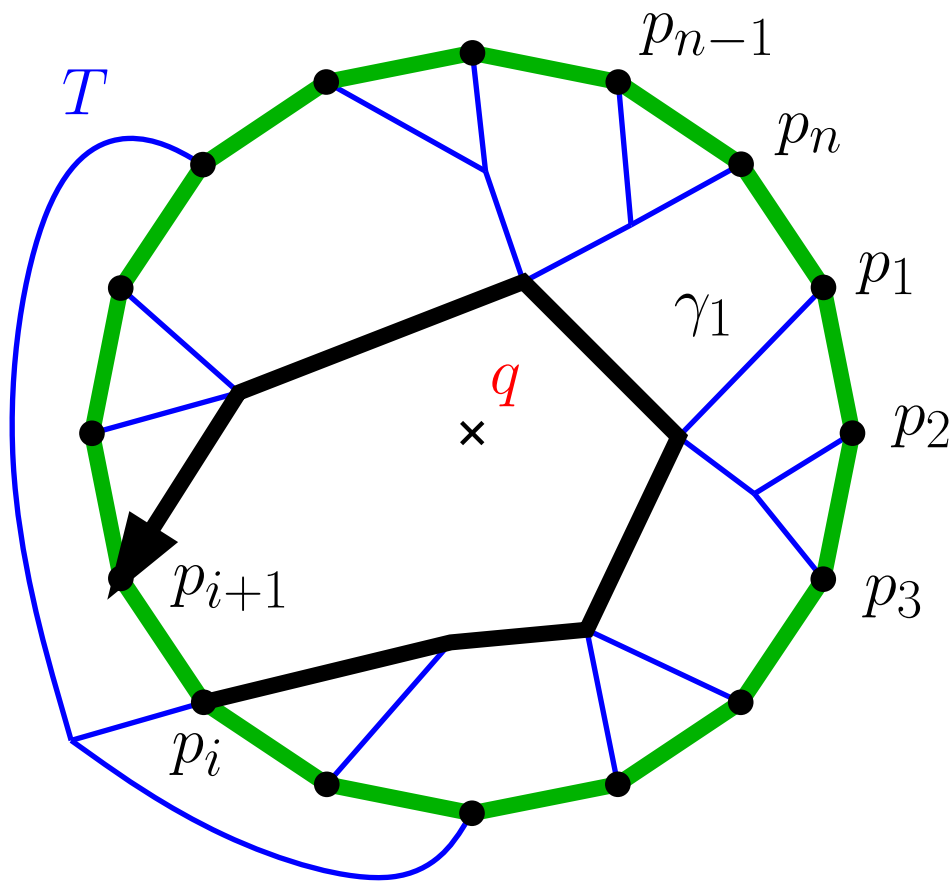
Let γ_i be path from p_i to p_{i+1} in T .

The concatenation of $\gamma_1\gamma_2\dots\gamma_n$ is also a loop. It is contractible in $R^2 \setminus \{q\}$.

Dilation $O(\frac{n}{k+1})$ is optimal in the worst case — case $k = 0$

Let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.



Proof: The loop $p_1p_2p_3\dots p_np_1$ is not contractible in $R^2 \setminus \{q\}$.

Let γ_i be path from p_i to p_{i+1} in T .

The concatenation of $\gamma_1\gamma_2\dots\gamma_n$ is also a loop. It is contractible in $R^2 \setminus \{q\}$.

→ There is an i such that

γ_i is not equivalent to segment p_ip_{i+1} .

The length of p_ip_{i+1} is about $2\pi/n$.

The length of γ_i is at least 2.

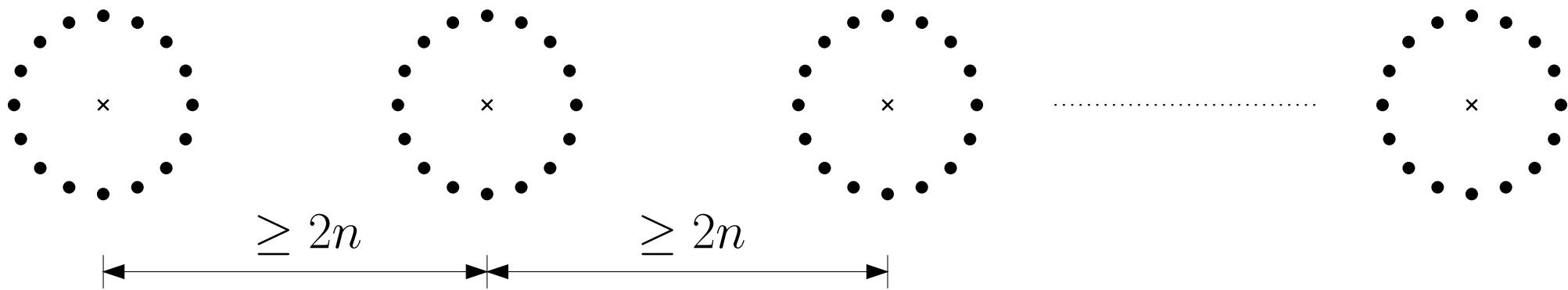
$$\rightarrow \text{dil}(T) \geq \frac{d_T(p_i, p_{i+1})}{\|p_ip_{i+1}\|} \geq \frac{2}{2\pi/n} = \frac{n}{\pi}.$$

Dilation $O(\frac{n}{k+1})$ is optimal in the worst case

$k = 0$: let S be n points spaced equally on the boundary of a unit circle, and let T be any tree whose vertex set includes S .

Then $\text{dil}(T) \geq n/\pi$.

$k > 0$: take $k + 1$ unit circles, with interdistance $\geq 2n$, and $\frac{n}{k+1}$ points each:



$n - 1 + k$ edges, $\geq k$ edges between circles; $< n$ edges within circles.

$k + 1$ circles \rightarrow at least one circle has less than $\frac{n}{k+1}$ edges;

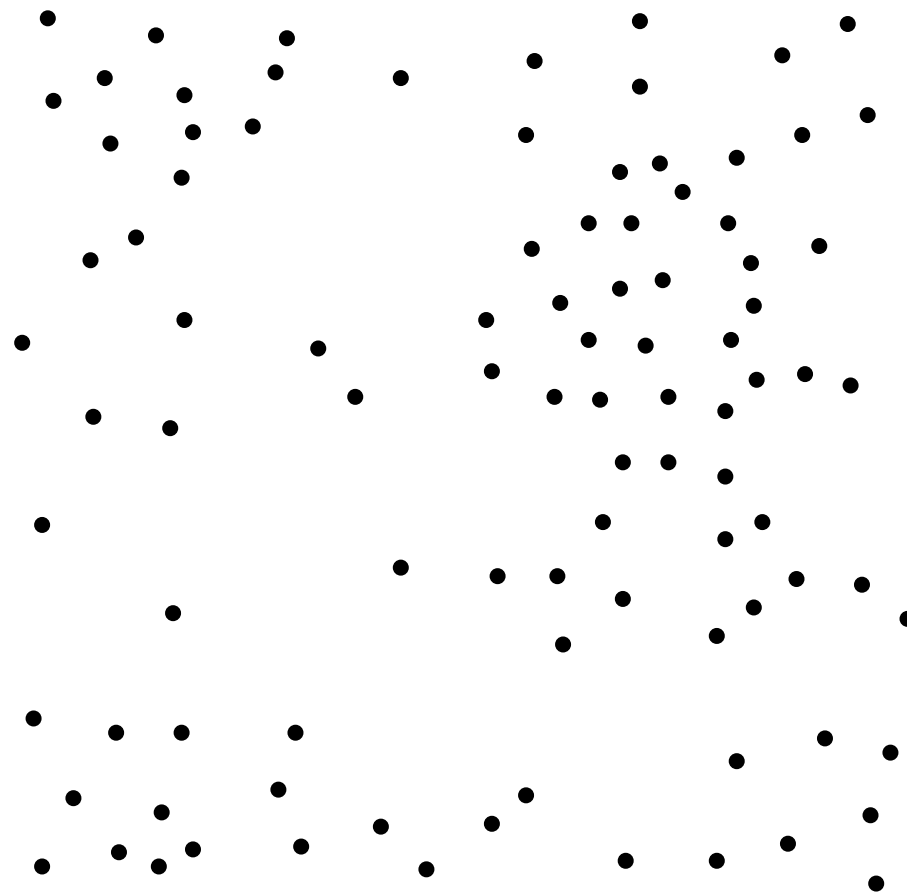
apply case $k = 0$ to that circle: $\text{dil}(T) \geq \frac{n}{k+1}/\pi$.

$n, k = |\text{points}|, |\text{added edges}|$; $d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T$; $\text{dil}_T(p, q) = d_T(p, q)/\|p, q\|$

Spread s : dilation $O(s/\sqrt{k+1})$

Spread $s(\text{set of points } S) = \frac{\text{longest pairwise distance in } S}{\text{shortest pairwise distance in } S}$

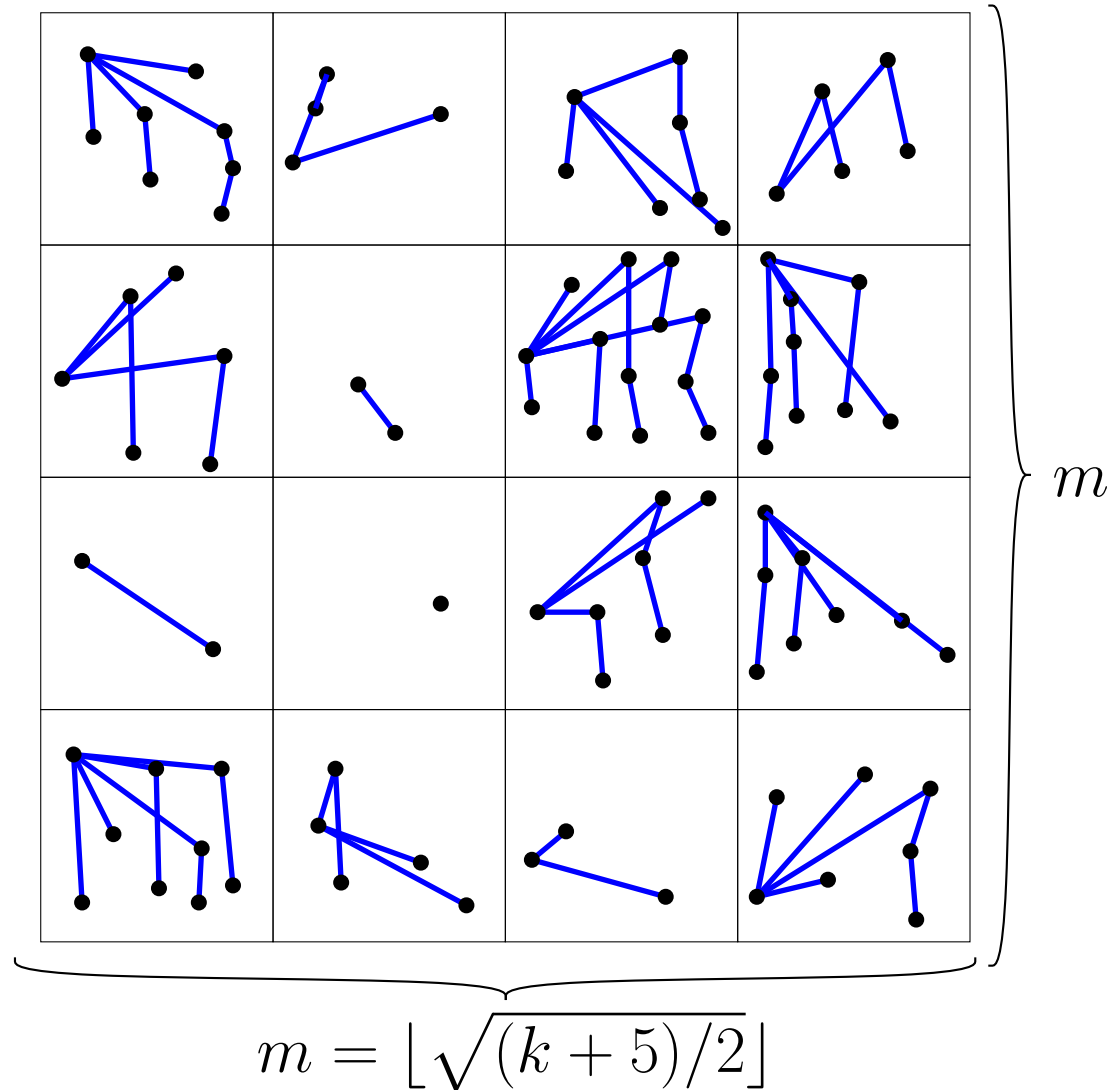
For any set S of n points
in the plane and any k
such that $s(S) = O(n/\sqrt{k})$
we can construct a spanner
on S with $n - 1 + k$ edges
and dilation $O(s(S)/\sqrt{k+1})$



Spread s : dilation $O(s/\sqrt{k+1})$

$$\text{Spread } s(\text{set of points } S) = \frac{\text{longest pairwise distance in } S}{\text{shortest pairwise distance in } S}$$

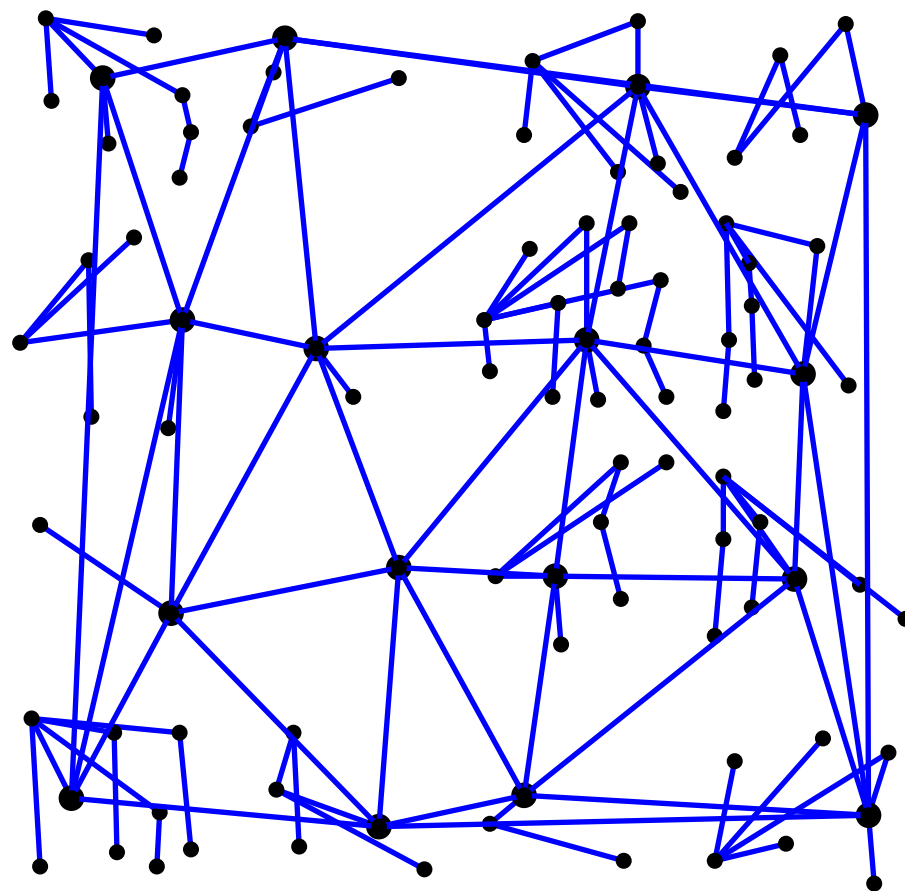
For any set S of n points in the plane and any k such that $s(S) = O(n/\sqrt{k})$ we can construct a spanner on S with $n - 1 + k$ edges and dilation $O(s(S)/\sqrt{k+1})$



Spread s : dilation $O(s/\sqrt{k+1})$

$$\text{Spread } s(\text{set of points } S) = \frac{\text{longest pairwise distance in } S}{\text{shortest pairwise distance in } S}$$

For any set S of n points
in the plane and any k
such that $s(S) = O(n/\sqrt{k})$
we can construct a spanner
on S with $n - 1 + k$ edges
and dilation $O(s(S)/\sqrt{k+1})$



$$n, k = |\text{points}|, |\text{added edges}|; \quad d_T(p, q) = \text{dist. } p \rightarrow q \text{ in } T; \quad \text{dil}_T(p, q) = d_T(p, q)/\|p, q\|$$

Spread s : dilation $\Omega(s/\sqrt{k+1})$

$$\text{Spread } s(\text{set of points } S) = \frac{\text{longest pairwise distance in } S}{\text{shortest pairwise distance in } S}$$

For any n, k, s ($s = O(n/\sqrt{k})$)

there is a set S of n points

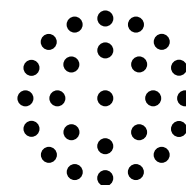
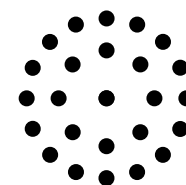
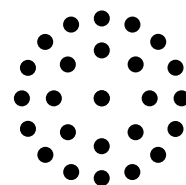
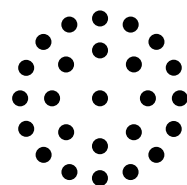
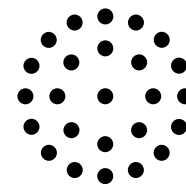
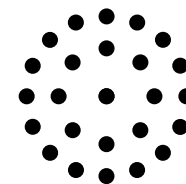
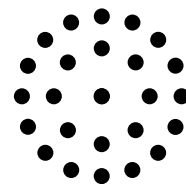
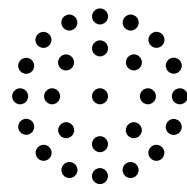
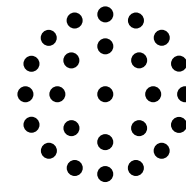
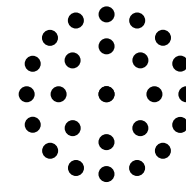
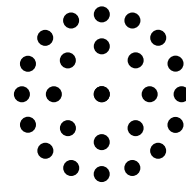
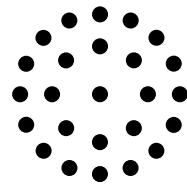
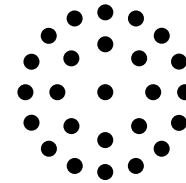
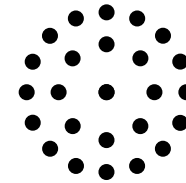
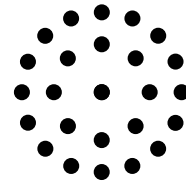
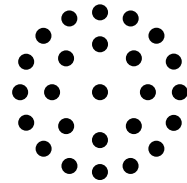
in the plane

with spread s such that

any spanner on S

with $n - 1 + k$ edges

has dilation $\Omega(s/\sqrt{k+1})$.



Conclusions

Results generalize to higher dimensions

(use t -spanner by Das and Heffernan instead of Delaunay triangulation)

Given k and a set of points S in R^d , ...

- our $O(n \log n)$ -time algorithm finds a spanner on S with at most $n - 1 + k$ edges and dilation $O(\frac{n}{k+1})$
- In the worst case, any graph with $n - 1 + k$ edges on S has dilation at least $\frac{2n}{\pi(k+1)}$.

Open problems:

- Find a spanner with *minimum* dilation. **Is the problem NP-complete?**
- Special case: points in plane, $k = 0$. Find minimum-dilation spanning *tree*. **Can it self-intersect?**

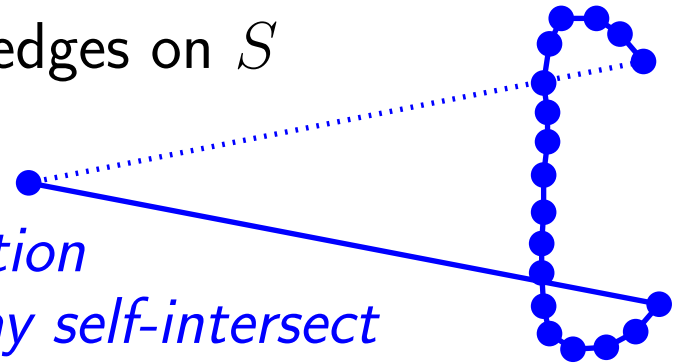
Conclusions

Results generalize to higher dimensions

(use t -spanner by Das and Heffernan instead of Delaunay triangulation)

Given k and a set of points S in R^d , ...

- our $O(n \log n)$ -time algorithm finds a spanner on S with at most $n - 1 + k$ edges and dilation $O(\frac{n}{k+1})$
- In the worst case, any graph with $n - 1 + k$ edges on S has dilation at least $\frac{2n}{\pi(k+1)}$.



A minimum-dilation path or cycle may self-intersect

Open problems:

- Find a spanner with *minimum* dilation. **Is the problem NP-complete?**
- Special case: points in plane, $k = 0$. Find minimum-dilation spanning *tree*. **Can it self-intersect?**

Conclusions

Results generalize to higher dimensions

(use t -spanner by Das and Heffernan instead of Delaunay)

Given k and a set of points S in R^d , ...

- our $O(n \log n)$ -time algorithm finds a spanner on S with at most $n - 1 + k$ edges and dilation $O(\frac{n}{k+1})$
- In the worst case, any graph with $n - 1 + k$ edges on S has dilation at least $\frac{2n}{\pi(k+1)}$.

Open problems:

- Find a spanner with *minimum* dilation. **Is the problem NP-complete?**
- Special case: points in plane, $k = 0$. Find minimum-dilation spanning *tree*.

Can it self-intersect?

**THAT'S ALL
FOLKS**

*A minimum-dilation
path or cycle may self-intersect*

