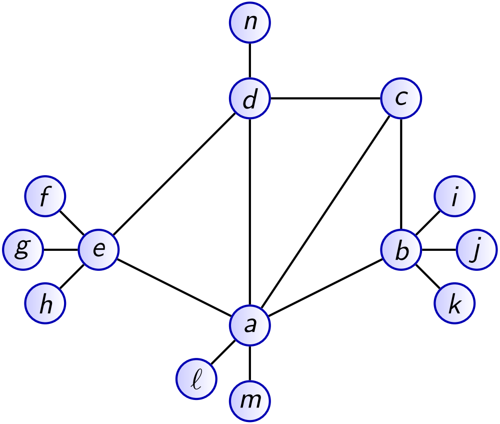
Graph Reconstruction via Distance Oracles

Claire Mathieu and Hang Zhou

École Normale Supérieure de Paris, France

July 9, 2013

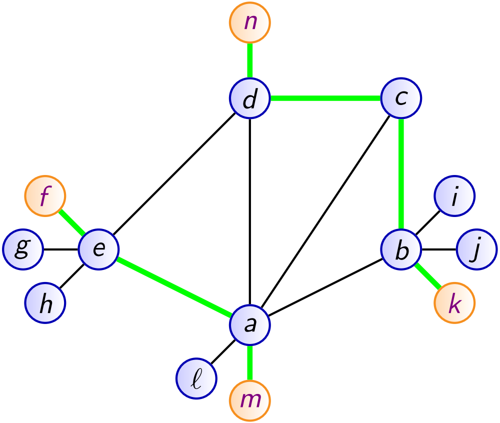
Network tomography and graph reconstruction



traceroute(n,k): $n - \star - \star - \star - k$

traceroute(f,m): $f - \star - \star - m$

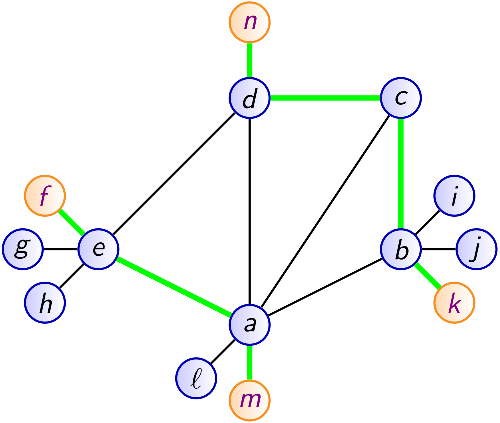
Network tomography and graph reconstruction



traceroute(n,k): $n - \star - \star - \star - k$

traceroute(f,m): $f - \star - \star - m$

Network tomography and graph reconstruction

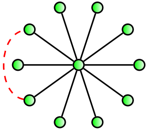


- Hidden: $G = (V, E)$
- QUERY oracle:
  $\{u, v\} \mapsto$ distance d_{uv}
- Output E
- Complexity: # queries

traceroute(n,k) : $n - \star - \star - \star - k$
traceroute(f,m): $f - \star - \star - m$

Results on general graphs

- $O(n^2)$ algorithm: Query every $\{u, v\}$. Output $\{\{u, v\} : d_{uv} = 1\}$
- $\Omega(n^2)$ lower bound [Reyzin Srivastava 07]



- Randomized $\tilde{O}(n^2/f)$ algorithm for approximate reconstruction:

$$\widehat{d_{uv}} \leq d_{uv} \leq f \cdot \widehat{d_{uv}}, \text{ for every } u \text{ and } v$$

- $\Omega(n^2/f)$ lower bound for approximate reconstruction

Main results

Graphs of bounded degree

Randomized $\tilde{O}(n\sqrt{n})$ algorithm

Outerplanar graphs of bounded degree

Randomized $\tilde{O}(n)$ algorithm

Main results

Graphs of bounded degree

Randomized $\tilde{O}(n\sqrt{n})$ algorithm

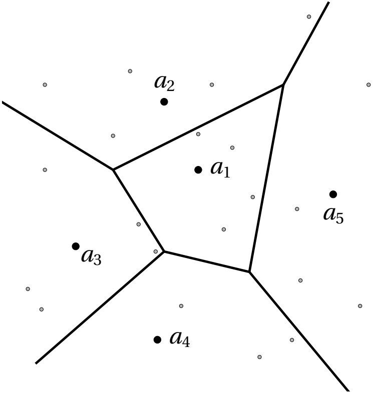
Outerplanar graphs of bounded degree

Randomized $\tilde{O}(n)$ algorithm

Graphs of bounded degree

A : set of centers

$Voronoi_A(a)$: set of vertices closer to a than to $A \backslash \{a\}$



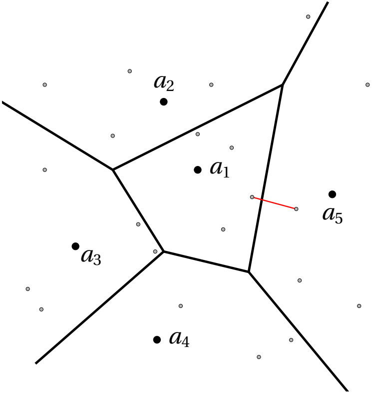
Defective Algorithm

1. Construct A
2. Reconstruct $G[Voronoi_A(a)]$ for all $a \in A$
3. Return $\bigcup_A G[Voronoi_A(a)]$

Graphs of bounded degree

A : set of centers

$Voronoi_A(a)$: set of vertices closer to a than to $A \backslash \{a\}$



Defective Algorithm

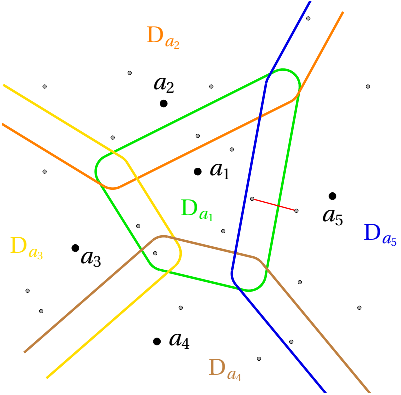
1. Construct A
2. Reconstruct $G[Voronoi_A(a)]$ for all $a \in A$
3. Return $\bigcup_A G[Voronoi_A(a)]$

Graphs of bounded degree: $\tilde{O}(n\sqrt{n})$ algorithm

A : set of centers

D_a : region associated to $a \in A$ including $Voronoi_A(a)$

Goal : $\bigcup_A G[D_a]$ covers E



Main Algorithm

1. Construct A of expected size $\tilde{O}(\sqrt{n})$ such that $\max_A |D_a| = O(\sqrt{n})$
2. For every $a \in A$, reconstruct $G[D_a]$ by querying $D_a \times D_a$
3. Return $\bigcup_a G[D_a]$

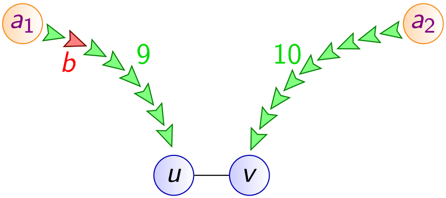
Graphs of bounded degree: $\tilde{O}(n\sqrt{n})$ algorithm

Definition: $D_a = \bigcup_b Voronoi_{A\cup\{b\}}(b)$.

$\Uparrow$

a, neighbor of a, or neighbor of neighbor of a

Lemma: $\bigcup_A G[D_a]$ covers E.



Main Algorithm

1. Construct A of expected size $\tilde{O}(\sqrt{n})$ such that $\max_A |D_a| = O(\sqrt{n})$
2. For every $a \in A$, reconstruct $G[D_a]$ by querying $D_a \times D_a$
3. Return $\bigcup_a G[D_a]$

$\tilde{O}(n\sqrt{n})$ algorithm: how to construct A

Idea [Thorup Zwick 2001]: keep adding to A
random vertices b such that $|Voronoi_{A\cup\{b\}}(b)| \geq \sqrt{n}$, until
every vertex b of V has $|Voronoi_{A\cup\{b\}}(b)| < \sqrt{n}$

Invariant: if $b \notin L$ then $|Voronoi_{A\cup\{b\}}(b)| < \sqrt{n}$

- $A \leftarrow \emptyset$ and $L \leftarrow V$
- While $L \neq \emptyset$:
  - $R \leftarrow$ sample from L of size $\tilde{O}(\sqrt{n})$
  - add R to A and query $R \times V$
  - for every $b \in L$, estimate $|Voronoi_{A\cup\{b\}}(b)|$ using $\tilde{O}\left(\sqrt{n}\right)$ queries
  - remove from L vertices with estimates less than $\sqrt{n}/10$

Query complexity: $(\#iterations) \times \tilde{O}\left(n\sqrt{n}\right) = \tilde{O}(n\sqrt{n})$

Main results

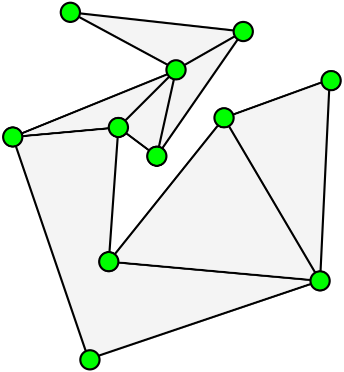
Graphs of bounded degree

Randomized $\tilde{O}(n\sqrt{n})$ algorithm

Outerplanar graphs of bounded degree

Randomized $\tilde{O}(n)$ algorithm

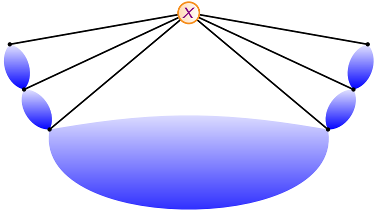
An outerplanar graph



Outerplanar graphs of bounded degree: $\tilde{O}(n)$ algorithm

Main Algorithm

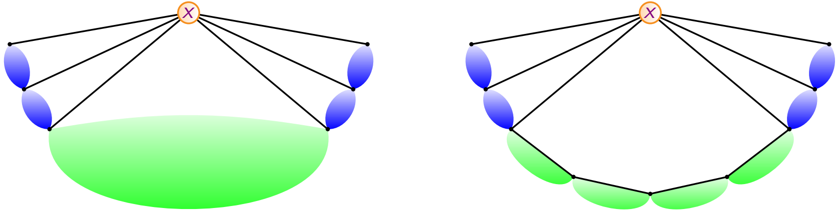
1. Use a well-chosen node x to decompose V into components
2. If a component has size $> 0.99n$, decompose it into subcomponents
3. Recurse on each piece



Outerplanar graphs of bounded degree: $\tilde{O}(n)$ algorithm

Main Algorithm

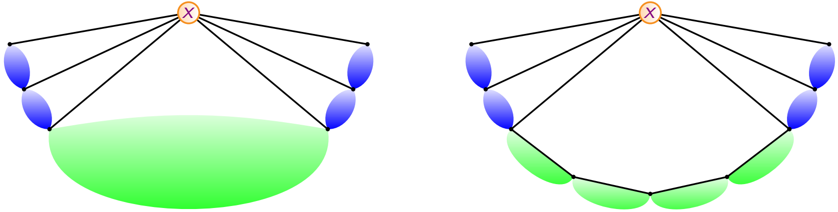
1. Use a well-chosen node x to decompose V into components
2. If a component has size $> 0.99n$, decompose it into subcomponents
3. Recurse on each piece



Outerplanar graphs of bounded degree: $\tilde{O}(n)$ algorithm

Main Algorithm

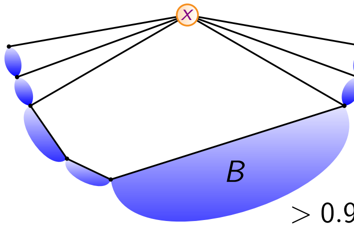
1. Use a well-chosen node x to decompose V into components
2. If a component has size $> 0.99n$, decompose it into subcomponents
3. Recurse on each piece



Why no subcomponent is huge ($> 0.99n$)?

Definition: x is well-chosen if at least $0.02n^2$ u-to-v shortest paths go through x.

Lemma: If x is well-chosen then no subcomponent is huge.



$> 0.98n^2$ pairs $\{u, v\}$ inside B

$\tilde{O}(n)$ queries for decomposition, then divide-and-conquer.

Above lemma $\implies$ $\tilde{O}(n)$ queries overall

How do we find a well-chosen vertex in $\tilde{O}(n)$ queries?

Definition: x is well-chosen if at least $0.02n^2$ u-to-v shortest paths go through x.

Fact: There exists some well-chosen vertex.

Algorithm to find a well-chosen vertex

- $R \leftarrow$ sample of $\log(n)$ pairs (u, v) from V^2
- For each $(u, v) \in R$, compute the u-to-v shortest paths
- Select the vertex that belongs to the most shortest paths

Thank you!