Lower Bounds for Graph Embeddings and Combinatorial Preconditioners

Peter Richter

Joint work with Gary Miller to appear in SPAA 2004

Introduction

A generic problem:

- How well can an n-node graph G be "approximated" by a spanning tree H?

Introduction

A generic problem:

- How well can an n-node graph G be "approximated" by a spanning tree H?

Three versions of the problem:

- Minimize the congestion $c(G, H)$, dilation $d(G, H)$, or condition number $\kappa_f(G, H)$

Introduction

A generic problem:

- How well can an n-node graph G be "approximated" by a spanning tree H?

Three versions of the problem:

- Minimize the congestion $c(G, H)$, dilation $d(G, H)$, or condition number $\kappa_f(G, H)$
- Applications: packet routing, linear systems

Introduction

A generic problem:

- How well can an n-node graph G be "approximated" by a spanning tree H?

Three versions of the problem:

- Minimize the congestion $c(G, H)$, dilation $d(G, H)$, or condition number $\kappa_f(G, H)$
- Applications: packet routing, linear systems
- Upper bounds: $c(G, H), d(G, H) \leq O(n)$ and $\kappa_f(G, H) \leq O(n^{1+o(1)})$ [Boman/Hendrickson]

Introduction

Folklore results:

- Congestion is largest when G has large separators: $c(G,H) \geq \Omega(n)$ when G is an expander

Introduction

Folklore results:

- Congestion is largest when G has large separators: $c(G,H) \geq \Omega(n)$ when G is an expander
- Dilation is largest when G has long cycles: $d(G,H) \geq \Omega(n)$ when G is a simple cycle

Introduction

Folklore results:

- Congestion is largest when G has large separators: $c(G, H) \geq \Omega(n)$ when G is an expander
- Dilation is largest when G has long cycles: $d(G, H) \geq \Omega(n)$ when G is a simple cycle

Main result:

- Condition number is largest when G is a square mesh: $\kappa_f(G, H) \geq \Omega(n^{1-o(1)})$

Outline

This talk:

- Introduction

Outline

This talk:

- Introduction
- Embeddings and preconditioners

Outline

This talk:

- Introduction
- Embeddings and preconditioners
- Expanders and cycles

Outline

This talk:

- Introduction
- Embeddings and preconditioners
- Expanders and cycles
- Square meshes

Outline

This talk:

- Introduction
- Embeddings and preconditioners
- Expanders and cycles
- Square meshes
- Conclusion

Embeddings and Preconditioners

A graph embedding $\varphi : G \hookrightarrow H$*...*

- Maps nodes in G to nodes in H

Embeddings and Preconditioners

A graph embedding $\varphi : G \hookrightarrow H$...

- Maps nodes in G to nodes in H
- Maps edges in G to paths in H

Embeddings and Preconditioners

A graph embedding $\varphi : G \hookrightarrow H$*...*

- Maps nodes in G to nodes in H
- Maps edges in G to paths in H

Measures of embedding quality:

- Congestion $c_\varphi(G, H)$: maximum number of H-paths in the image of φ sharing a single H-edge

Embeddings and Preconditioners

A graph embedding $\varphi : G \hookrightarrow H$*...*

- Maps nodes in G to nodes in H
- Maps edges in G to paths in H

Measures of embedding quality:

- Congestion $c_\varphi(G, H)$: maximum number of H-paths in the image of φ sharing a single H-edge
- Dilation $d_\varphi(G, H)$: maximum length of an H-path in the image of φ

Embeddings and Preconditioners

The routing view of an embedding:

- G is guest (demands), H is host (links)

Embeddings and Preconditioners

The routing view of an embedding:

- G is guest (demands), H is host (links)
- Congestion is bottleneck, dilation is delay

Embeddings and Preconditioners

The routing view of an embedding:

- G is guest (demands), H is host (links)
- Congestion is bottleneck, dilation is delay

The packet routing problem:

- Step 1: Path selection (fixed by embedding)

Embeddings and Preconditioners

The routing view of an embedding:

- G is guest (demands), H is host (links)
- Congestion is bottleneck, dilation is delay

The packet routing problem:

- Step 1: Path selection (fixed by embedding)
- Step 2: Motion schedule (model-dependent)

Embeddings and Preconditioners

The routing view of an embedding:

- G is guest (demands), H is host (links)
- Congestion is bottleneck, dilation is delay

The packet routing problem:

- Step 1: Path selection (fixed by embedding)
- Step 2: Motion schedule (model-dependent)
- Solvable in time $\Theta(c_\varphi(G,H) + d_\varphi(G,H))$ in a particular store-and-forward model [Leighton/Maggs/Rao]

Embeddings and Preconditioners

Laplacian matrices:

- Start with an n-node graph $G = (V, E)$

Embeddings and Preconditioners

Laplacian matrices:

- Start with an n-node graph $G = (V, E)$
- Form an $n \times n$ matrix: $G_{ij} = -1$ if $(i, j) \in E_G$, $G_{ij} = 0$ if $(i, j) \notin E_G$, and $G_{ii} = \mathrm{degree}(i)$

Embeddings and Preconditioners

Laplacian matrices:

- Start with an n-node graph $G = (V, E)$
- Form an $n \times n$ matrix: $G_{ij} = -1$ if $(i, j) \in E_G$, $G_{ij} = 0$ if $(i, j) \notin E_G$, and $G_{ii} = \mathrm{degree}(i)$
- G is symmetric positive semidefinite, with nullspace j (the all-ones vector)

Embeddings and Preconditioners

Laplacian matrices:

- Start with an n-node graph $G = (V, E)$
- Form an $n \times n$ matrix: $G_{ij} = -1$ if $(i, j) \in E_G$, $G_{ij} = 0$ if $(i, j) \notin E_G$, and $G_{ii} = \mathrm{degree}(i)$
- G is symmetric positive semidefinite, with nullspace j (the all-ones vector)

M*-matrices:*

- Generalization of Laplacian matrices

Embeddings and Preconditioners

Laplacian matrices:

- Start with an n-node graph $G = (V, E)$
- Form an $n \times n$ matrix: $G_{ij} = -1$ if $(i, j) \in E_G$, $G_{ij} = 0$ if $(i, j) \notin E_G$, and $G_{ii} = \mathrm{degree}(i)$
- G is symmetric positive semidefinite, with nullspace j (the all-ones vector)

M-matrices:

- Generalization of Laplacian matrices
- Arise in FDM/FEM for elliptic PDEs

Embeddings and Preconditioners

A generalized eigenvalue problem:

- $Gx = \lambda Hx$ where G,H are Laplacian

Embeddings and Preconditioners

A generalized eigenvalue problem:

- $Gx = \lambda Hx$ where G,H are Laplacian
- $\max \lambda_f(G, H)$: maximum λ with $x \perp j$

Embeddings and Preconditioners

A generalized eigenvalue problem:

- $Gx = \lambda Hx$ where G,H are Laplacian
- $\max \lambda_f(G, H)$: maximum λ with $x \perp j$
- $\min \lambda_f(G, H)$: minimum λ with $x \perp j$

Embeddings and Preconditioners

A generalized eigenvalue problem:

- $Gx = \lambda Hx$ where G,H are Laplacian
- $\max \lambda_f(G, H)$: maximum λ with $x \perp j$
- $\min \lambda_f(G, H)$: minimum λ with $x \perp j$

A generalized condition number:

- $\kappa_f(G, H) = \max_{x \perp j} \frac{x^T G x}{x^T H x} \cdot \max_{x \perp j} \frac{x^T H x}{x^T G x} = (\max \lambda_f(G, H)) \cdot (\min \lambda_f(G, H))^{-1}$

Embeddings and Preconditioners

Electrical network view:

- $G = (V, E)$ is a resistive circuit

Embeddings and Preconditioners

Electrical network view:

- $G = (V, E)$ is a resistive circuit
- Nodes are junctions/terminals

Embeddings and Preconditioners

Electrical network view:

- $G = (V, E)$ is a resistive circuit
- Nodes are junctions/terminals
- Edges are branch conductors

Embeddings and Preconditioners

Electrical network view:

- $G = (V, E)$ is a resistive circuit
- Nodes are junctions/terminals
- Edges are branch conductors

Energy interpretation:

- The power law:
  $\mathcal{E}_G(x) = x^T G x = \sum_{(i,j) \in E_G} (x_i - x_j)^2$

Embeddings and Preconditioners

Electrical network view:

- $G = (V, E)$ is a resistive circuit
- Nodes are junctions/terminals
- Edges are branch conductors

Energy interpretation:

- The power law:
  $\mathcal{E}_G(x) = x^T G x = \sum_{(i,j) \in E_G} (x_i - x_j)^2$
- Rayleigh quotients compare power dissepation

Embeddings and Preconditioners

The classical conjugate gradient method:

- Solve $Gx = b$ iteratively, using a preconditioner H

Embeddings and Preconditioners

The classical conjugate gradient method:

- Solve $Gx = b$ iteratively, using a preconditioner H
- Iteration cost: matrix-vector multiply involving G, direct system-solve involving H

Embeddings and Preconditioners

The classical conjugate gradient method:

- Solve $Gx = b$ iteratively, using a preconditioner H
- Iteration cost: matrix-vector multiply involving G, direct system-solve involving H
- Iteration count: $O(\sqrt{\kappa_f(G, H)})$

Embeddings and Preconditioners

The classical conjugate gradient method:

- Solve $Gx = b$ iteratively, using a preconditioner H
- Iteration cost: matrix-vector multiply involving G, direct system-solve involving H
- Iteration count: $O(\sqrt{\kappa_f(G, H)})$
- Tradeoff: H should be sparser than G but approximate G well

Embeddings and Preconditioners

The classical conjugate gradient method:

- Solve $Gx = b$ iteratively, using a preconditioner H
- Iteration cost: matrix-vector multiply involving G, direct system-solve involving H
- Iteration count: $O(\sqrt{\kappa_f(G, H)})$
- Tradeoff: H should be sparser than G but approximate G well
- Example: H is a spanning tree of G

Embeddings and Preconditioners

Bounding the condition number from above:

- $\kappa_f(G,H) \leq O(\min_\varphi c_\varphi(G,H) \cdot d_\varphi(G,H))$ [Gremban]

Embeddings and Preconditioners

Bounding the condition number from above:

- $\kappa_f(G,H) \leq O(\min_\varphi c_\varphi(G,H) \cdot d_\varphi(G,H))$ [Gremban]

Bounding the condition number from below:

- Congestion-times-dilation is strong, but false

Embeddings and Preconditioners

Bounding the condition number from above:

- $\kappa_f(G,H) \leq O(\min_\varphi c_\varphi(G,H) \cdot d_\varphi(G,H))$ [Gremban]

Bounding the condition number from below:

- Congestion-times-dilation is strong, but false
- Congestion-plus-dilation is easy, but weak

Embeddings and Preconditioners

Bounding the condition number from above:

- $\kappa_f(G,H) \le O(\min_\varphi c_\varphi(G,H) \cdot d_\varphi(G,H))$ [Gremban]

Bounding the condition number from below:

- Congestion-times-dilation is strong, but false
- Congestion-plus-dilation is easy, but weak
- For the simple square mesh, novel techniques are needed

Expanders and Cycles

Expander:

- bounded-degree, with linear-size separators

Expanders and Cycles

Expander:

- bounded-degree, with linear-size separators
- large congestion, small dilation, large condition number

Expanders and Cycles

Expander:

- bounded-degree, with linear-size separators
- large congestion, small dilation, large condition number

Simple cycle:

- connected, degree-two

Expanders and Cycles

Expander:

- bounded-degree, with linear-size separators
- large congestion, small dilation, large condition number

Simple cycle:

- connected, degree-two
- small congestion, large dilation, large condition number

Expanders and Cycles

Congestion for expander:

- Find single-edge separator of H into U, $V \setminus U$

Expanders and Cycles

Congestion for expander:

- Find single-edge separator of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(n)$

Expanders and Cycles

Congestion for expander:

- Find single-edge separator of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(n)$
- Hence, $c_\varphi(G, H) \geq \Omega(n)$ for any φ

Expanders and Cycles

Congestion for expander:

- Find single-edge separator of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(n)$
- Hence, $c_\varphi(G, H) \geq \Omega(n)$ for any φ

Condition number for expander:

- Set potential x at U, $V \setminus U$ to 0, 1

Expanders and Cycles

Congestion for expander:

- Find single-edge separator of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(n)$
- Hence, $c_\varphi(G, H) \geq \Omega(n)$ for any φ

Condition number for expander:

- Set potential x at U, $V \setminus U$ to 0, 1
- $\mathcal{E}_G(x)$, $\mathcal{E}_H(x)$ count separator edges

Expanders and Cycles

Congestion for expander:

- Find single-edge separator of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(n)$
- Hence, $c_\varphi(G, H) \geq \Omega(n)$ for any φ

Condition number for expander:

- Set potential x at U, $V \setminus U$ to 0, 1
- $\mathcal{E}_G(x)$, $\mathcal{E}_H(x)$ count separator edges
- Hence, $\kappa_f(G, H) \geq \Omega(n)$

Expanders and Cycles

Dilation for simple cycle:

- Find the edge $(i, j) \in E_G$ missing from H

Expanders and Cycles

Dilation for simple cycle:

- Find the edge $(i, j) \in E_G$ missing from H
- This edge must be mapped around the cycle

Expanders and Cycles

Dilation for simple cycle:

- Find the edge $(i,j) \in E_G$ missing from H
- This edge must be mapped around the cycle
- Hence, $d_\varphi(G,H) \geq \Omega(n)$ for any φ

Expanders and Cycles

Dilation for simple cycle:

- Find the edge $(i, j) \in E_G$ missing from H
- This edge must be mapped around the cycle
- Hence, $d_\varphi(G, H) \geq \Omega(n)$ for any φ

Condition number for simple cycle:

- Set potential x at $i, \ldots, j$ to $0, \ldots, n$

Expanders and Cycles

Dilation for simple cycle:

- Find the edge $(i, j) \in E_G$ missing from H
- This edge must be mapped around the cycle
- Hence, $d_\varphi(G, H) \geq \Omega(n)$ for any φ

Condition number for simple cycle:

- Set potential x at $i, \ldots, j$ to $0, \ldots, n$
- $\mathcal{E}_G(x) \geq n^2$, $\mathcal{E}_H(x) \leq O(n)$

Expanders and Cycles

Dilation for simple cycle:

- Find the edge $(i, j) \in E_G$ missing from H
- This edge must be mapped around the cycle
- Hence, $d_\varphi(G, H) \geq \Omega(n)$ for any φ

Condition number for simple cycle:

- Set potential x at $i, \ldots, j$ to $0, \ldots, n$
- $\mathcal{E}_G(x) \geq n^2$, $\mathcal{E}_H(x) \leq O(n)$
- Hence, $\kappa_f(G, H) \geq \Omega(n)$

Square Meshes

Square mesh:

- Planar (product of lines) or toroidal (product of cycles)

Square Meshes

Square mesh:

- Planar (product of lines) or toroidal (product of cycles)
- Two-dimensional (not an extreme case)

Square Meshes

Square mesh:

- Planar (product of lines) or toroidal (product of cycles)
- Two-dimensional (not an extreme case)
- Common in practice (e.g., in FDM/FEM)

Square Meshes

Square mesh:

- Planar (product of lines) or toroidal (product of cycles)
- Two-dimensional (not an extreme case)
- Common in practice (e.g., in FDM/FEM)
- Medium congestion, medium dilation, large condition number

Square Meshes

Congestion for square mesh:

- Find one-edge separator e of H into U, $V \setminus U$

Square Meshes

Congestion for square mesh:

- Find one-edge separator e of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(\sqrt{n})$

Square Meshes

Congestion for square mesh:

- Find one-edge separator e of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(\sqrt{n})$
- Hence, $c_\varphi(G, H) \geq \Omega(\sqrt{n})$ for any φ

Square Meshes

Congestion for square mesh:

- Find one-edge separator e of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(\sqrt{n})$
- Hence, $c_\varphi(G, H) \geq \Omega(\sqrt{n})$ for any φ

Dilation for square mesh:

- Separator of G into U, $V \setminus U$ contains at least one edge at distance $\Omega(\sqrt{n})$ from e

Square Meshes

Congestion for square mesh:

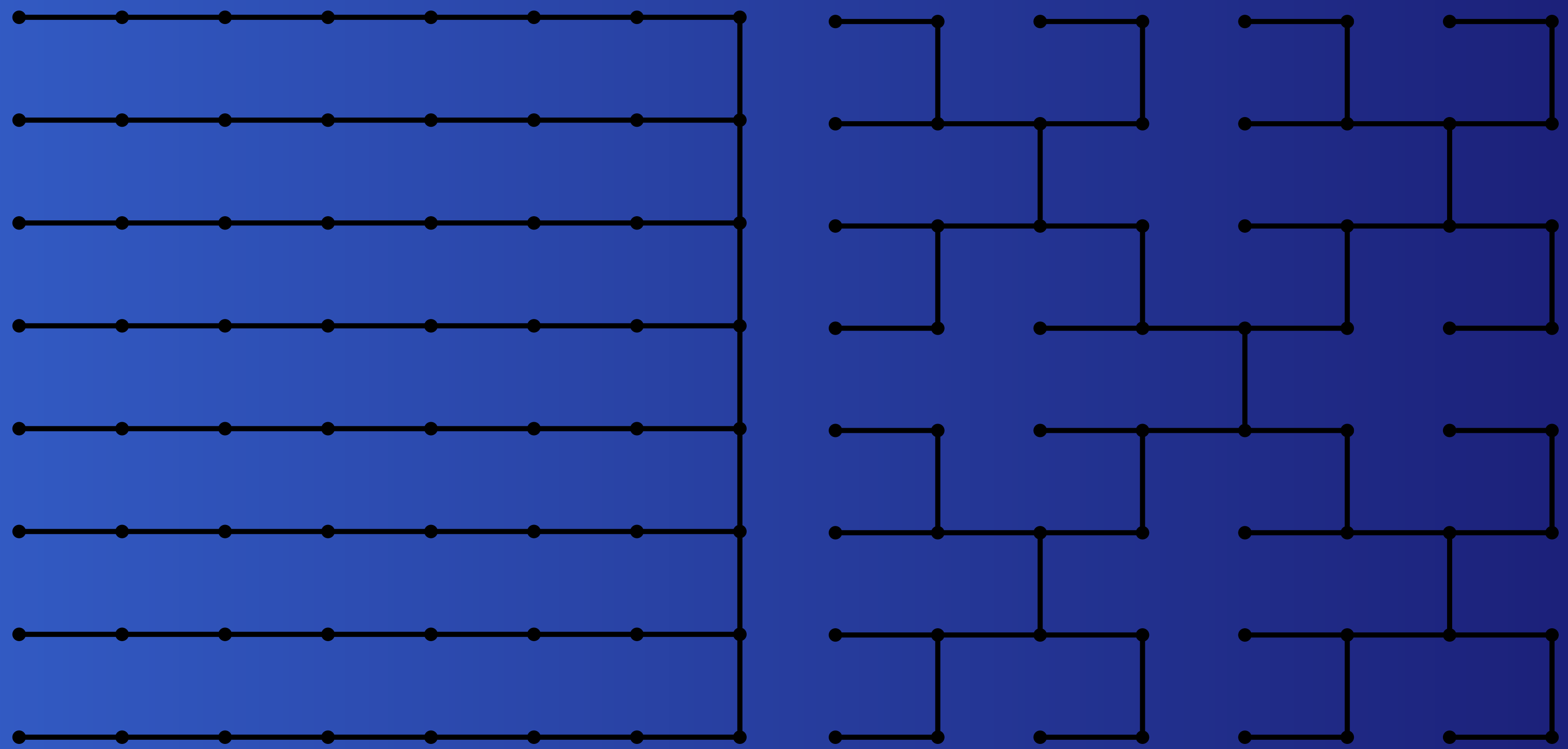
- Find one-edge separator e of H into U, $V \setminus U$
- Separator of G into U, $V \setminus U$ has size $\Omega(\sqrt{n})$
- Hence, $c_\varphi(G, H) \geq \Omega(\sqrt{n})$ for any φ

Dilation for square mesh:

- Separator of G into U, $V \setminus U$ contains at least one edge at distance $\Omega(\sqrt{n})$ from e
- Hence, $d_\varphi(G, H) \geq \Omega(\sqrt{n})$ for any φ

Square Meshes

Two spanning trees for the planar square mesh:



Square Meshes

Upper bounds for the planar square mesh:

- Let H be either of the spanning trees from the previous slide

Square Meshes

Upper bounds for the planar square mesh:

- Let H be either of the spanning trees from the previous slide
- Let φ be the shortest-path map

Square Meshes

Upper bounds for the planar square mesh:

- Let H be either of the spanning trees from the previous slide
- Let φ be the shortest-path map
- Clear that $c_\varphi(G, H), d_\varphi(G, H) \leq O(\sqrt{n})$

Square Meshes

Upper bounds for the planar square mesh:

- Let H be either of the spanning trees from the previous slide
- Let φ be the shortest-path map
- Clear that $c_\varphi(G,H), d_\varphi(G,H) \leq O(\sqrt{n})$
- Hence, $\kappa_f(G,H) \leq O(n)$

Square Meshes

Lower bounds for particular spanning trees:

- Let H be either of the spanning trees from the previous slide

Square Meshes

Lower bounds for particular spanning trees:

- Let H be either of the spanning trees from the previous slide
- Congestion argument: $\kappa_f(G,H) \geq \Omega(\sqrt{n})$

Square Meshes

Lower bounds for particular spanning trees:

- Let H be either of the spanning trees from the previous slide
- Congestion argument: $\kappa_f(G,H) \geq \Omega(\sqrt{n})$
- Dilation argument: $\kappa_f(G,H) \geq \Omega(\sqrt{n})$

Square Meshes

Lower bounds for particular spanning trees:

- Let H be either of the spanning trees from the previous slide
- Congestion argument: $\kappa_f(G, H) \geq \Omega(\sqrt{n})$
- Dilation argument: $\kappa_f(G, H) \geq \Omega(\sqrt{n})$
- Hybrid argument: $\kappa_f(G, H) \geq \Omega(n)$

Square Meshes

Lower bounds for particular spanning trees:

- Let H be either of the spanning trees from the previous slide
- Congestion argument: $\kappa_f(G, H) \geq \Omega(\sqrt{n})$
- Dilation argument: $\kappa_f(G, H) \geq \Omega(\sqrt{n})$
- Hybrid argument: $\kappa_f(G, H) \geq \Omega(n)$
- Is there a better spanning tree?

Square Meshes

Main result:

- If G is a square mesh and H is a spanning tree, then $\kappa_f(G,H) \geq \Omega(n^{1-o(1)})$

Square Meshes

Main result:

- If G is a square mesh and H is a spanning tree, then $\kappa_f(G, H) \geq \Omega(n^{1-o(1)})$

Proof idea:

- Decompose H into subtrees, recursively

Square Meshes

Main result:

- If G is a square mesh and H is a spanning tree, then $\kappa_f(G,H) \geq \Omega(n^{1-o(1)})$

Proof idea:

- Decompose H into subtrees, recursively
- Look at the shapes of the subtrees

Square Meshes

Main result:

- If G is a square mesh and H is a spanning tree, then $\kappa_f(G, H) \geq \Omega(n^{1-o(1)})$

Proof idea:

- Decompose H into subtrees, recursively
- Look at the shapes of the subtrees
- Bound $\kappa_f(G, H)$ by finding a particular shape

Square Meshes

A tree decomposition:

- Let subtree S have mesh-diameter d

Square Meshes

A tree decomposition:

- Let subtree S have mesh-diameter d
- Let P be a path in S of mesh-diameter d between $H \setminus S$ and a leaf of S

Square Meshes

A tree decomposition:

- Let subtree S have mesh-diameter d
- Let P be a path in S of mesh-diameter d between $H \setminus S$ and a leaf of S
- Divide P into d/s subintervals of mesh-diameter $s << d$

Square Meshes

A tree decomposition:

- Let subtree S have mesh-diameter d
- Let P be a path in S of mesh-diameter d between $H \setminus S$ and a leaf of S
- Divide P into d/s subintervals of mesh-diameter $s << d$
- This partitions S into d/s subtrees of mesh-diameter s

Square Meshes

A tree decomposition:

- Let subtree S have mesh-diameter d
- Let P be a path in S of mesh-diameter d between $H \setminus S$ and a leaf of S
- Divide P into d/s subintervals of mesh-diameter $s << d$
- This partitions S into d/s subtrees of mesh-diameter s
- Either some tree is ill-shaped, or none are

Square Meshes

Lemma A (for well-shaped subtrees):

- Suppose edges e, f cut subtree S' into connected components L, R, C

Square Meshes

Lemma A (for well-shaped subtrees):

- Suppose edges e, f cut subtree S' into connected components L, R, C
- Let G have p edges from L to R, and let the C-path from e to f have length q

Square Meshes

Lemma A (for well-shaped subtrees):

- Suppose edges e, f cut subtree S' into connected components L, R, C
- Let G have p edges from L to R, and let the C-path from e to f have length q
- Then $\kappa_f(G, H) \geq \Omega(pq)$

Square Meshes

Lemma A (for well-shaped subtrees):

- Suppose edges e, f cut subtree S' into connected components L, R, C
- Let G have p edges from L to R, and let the C-path from e to f have length q
- Then $\kappa_f(G, H) \geq \Omega(pq)$
- Proof: Set potential x at $e, \ldots, f$ to $0, \ldots, q$; then $\mathcal{E}_G(x) \geq \Omega(pq^2)$ and $\mathcal{E}_H(x) \leq O(q)$

Square Meshes

Lemma A (for well-shaped subtrees):

- Suppose edges e, f cut subtree S' into connected components L, R, C
- Let G have p edges from L to R, and let the C-path from e to f have length q
- Then $\kappa_f(G, H) \geq \Omega(pq)$
- Proof: Set potential x at $e, \ldots, f$ to $0, \ldots, q$; then $\mathcal{E}_G(x) \geq \Omega(pq^2)$ and $\mathcal{E}_H(x) \leq O(q)$
- Example: $p, q \geq \Omega(\sqrt{n}) \Rightarrow \kappa_f(G, H) \geq \Omega(n)$

Square Meshes

Lemma B (for ill-shaped subtrees):

- Suppose subtree S' has diameter t and size r

Square Meshes

Lemma B (for ill-shaped subtrees):

- Suppose subtree S' has diameter t and size r
- Then $\kappa_f(G, H) \geq \Omega(t^3/r)$

Square Meshes

Lemma B (for ill-shaped subtrees):

- Suppose subtree S' has diameter t and size r
- Then $\kappa_f(G, H) \geq \Omega(t^3/r)$
- Proof: Choose a node u at one end of S', and set potential x at each $v \in E_{S'}$ to (u, v) distance; then $\mathcal{E}_G(x) \geq \Omega(t^3)$ and $\mathcal{E}_H(x) \leq O(r)$

Square Meshes

Lemma B (for ill-shaped subtrees):

- Suppose subtree S' has diameter t and size r
- Then $\kappa_f(G, H) \geq \Omega(t^3/r)$
- Proof: Choose a node u at one end of S', and set potential x at each $v \in E_{S'}$ to (u, v) distance; then $\mathcal{E}_G(x) \geq \Omega(t^3)$ and $\mathcal{E}_H(x) \leq O(r)$
- Example: $t \geq \Omega(\sqrt{n}), r \leq O(\sqrt{n}) \Rightarrow \kappa_f(G, H) \geq \Omega(n)$

Square Meshes

Proof of theorem:

- Fix $\epsilon > 0$, choose $s_1(\epsilon) << d = \Theta(\sqrt{n})$

Square Meshes

Proof of theorem:

- Fix $\epsilon > 0$, choose $s_1(\epsilon) << d = \Theta(\sqrt{n})$
- Start with a subtree S of mesh-diameter d, and perform a tree decomposition with parameter $s_1(\epsilon)$

Square Meshes

Proof of theorem:

- Fix $\epsilon > 0$, choose $s_1(\epsilon) << d = \Theta(\sqrt{n})$
- Start with a subtree S of mesh-diameter d, and perform a tree decomposition with parameter $s_1(\epsilon)$
- If no subtree is ill-shaped, apply Lemma A to conclude that $\kappa_f(G, H) \geq \Omega(n^{1-o(1)})$

Square Meshes

Proof of theorem (cont'd):

- If some subtree is ill-shaped, recurse; i.e., perform a tree decomposition on it with parameter $s_2(\epsilon) < s_1(\epsilon)$

Square Meshes

Proof of theorem (cont'd):

- If some subtree is ill-shaped, recurse; i.e., perform a tree decomposition on it with parameter $s_2(\epsilon) < s_1(\epsilon)$
- Repeat as necessary until some subtree is extremely ill-shaped, then apply Lemma B to conclude that $\kappa_f(G, H) \geq \Omega(n^{1-o(1)})$

Conclusion

Extension to spanning subgraphs:

- Let H have Euler characteristic k

Conclusion

Extension to spanning subgraphs:

- Let H have Euler characteristic k
- Partition H into “vines”

Conclusion

Extension to spanning subgraphs:

- Let H have Euler characteristic k
- Partition H into “vines”
- Lower bounds hold with n replaced by $\frac{n}{k+1}$

Conclusion

Extension to spanning subgraphs:

- Let H have Euler characteristic k
- Partition H into “vines”
- Lower bounds hold with n replaced by $\frac{n}{k+1}$
- Upper bounds hold similarly [Spielman/Teng]

Conclusion

Open questions:

- Is $\kappa_f(G, H) = \Theta(n)$ optimal for the square mesh?

Conclusion

Open questions:

- Is $\kappa_f(G, H) = \Theta(n)$ optimal for the square mesh?
- Is there a single spanning tree optimizing congestion, dilation, and condition number simultaneously?

Conclusion

Open questions:

- Is $\kappa_f(G, H) = \Theta(n)$ optimal for the square mesh?
- Is there a single spanning tree optimizing congestion, dilation, and condition number simultaneously?
- Can we find the optimal spanning tree efficiently?