Extensions and Limits to Vertex Sparsification

Ankur Moitra, MIT
joint work with Tom Leighton

June 5, 2010
Flow Sparsification

$G = (V, E)$
Flow Sparsification

\[ G = (V, E) \]
Flow Sparsification

\[ G = (V, E) \]

\[ K = \{a, b, c, d\} \]
Flow Sparsification

\[ G = (V, E) \]

\[ K = \{a, b, c, d\} \]
Flow Sparsification

$G = (V, E)$

$H = (K, E_H)$

$K = \{a, b, c, d\}$
Multicommodity Flow and Congestion
Multicommodity Flow and Congestion
Multicommodity Flow and Congestion
Multicommodity Flow and Congestion

\[
\begin{pmatrix}
0 \\
0 \\
x \\
y \\
0 \\
0
\end{pmatrix}
\]

\[
\begin{align*}
a &\rightarrow b \\
a &\rightarrow c \\
a &\rightarrow d \\
b &\rightarrow c \\
b &\rightarrow d \\
c &\rightarrow d
\end{align*}
\]
Multicommodity Flow and Congestion

\[ \begin{bmatrix} 0 \\ 0 \\ x \\ y \\ 0 \\ 0 \end{bmatrix} \]

\[ \begin{align*} &a \leftarrow b \\ &a \leftarrow c \\ &a \leftarrow d \\ &b \leftarrow c \\ &b \leftarrow d \\ &c \leftarrow d \end{align*} \]
Multicommodity Flow and Congestion

\[ f = \begin{bmatrix} 0 \\ 0 \\ x \\ y \\ 0 \\ 0 \end{bmatrix} \]

\[
\begin{align*}
\text{a} & \rightarrow \text{b} \\
\text{a} & \rightarrow \text{c} \\
\text{a} & \rightarrow \text{d} \\
\text{b} & \rightarrow \text{c} \\
\text{b} & \rightarrow \text{d} \\
\text{c} & \rightarrow \text{d} \\
\end{align*}
\]
Multicommodity Flow and Congestion

\[ f = \begin{bmatrix} 0 \\ 0 \\ x \\ y \\ 0 \\ 0 \end{bmatrix} \]

Graph G with nodes a, b, c, d, and edges x, y, e.
Multicommodity Flow and Congestion

\[
\text{cong}(\text{routing}) = \max_e \frac{\text{flow}(e)}{\text{capacity}(e)}
\]

\[
\begin{bmatrix}
0 \\
0 \\
x \\
y \\
0 \\
0
\end{bmatrix}
\]

\[
\begin{align*}
\vec{f} &= a \rightarrow b \\
&\quad a \rightarrow c \\
&\quad x \\
&\quad a \rightarrow d \\
&\quad y \\
&\quad b \rightarrow c \\
&\quad b \rightarrow d \\
&\quad c \rightarrow d
\end{align*}
\]
Multicommodity Flow and Congestion

\[ \text{cong}_G(f) = \min_{\text{routings}} \max_e \frac{\text{flow}(e)}{\text{capacity}(e)} \]
Flow Sparsification

\[ G = (V, E) \]

\[ K = \{a, b, c, d\} \]

\[ H = (K, E_H) \]
A graph $H = (K, E_H)$ on just the terminal set is a **Flow-Sparsifier** if for all demands $\vec{f} \in \mathcal{R}^{(K)}_2$

$$cong_H(\vec{f}) \leq cong_G(\vec{f})$$
Definition

A graph $H = (K, E_H)$ on just the terminal set is a **Flow-Sparsifier** if for all demands $\vec{f} \in \mathcal{R}^{(K)}$

$$cong_H(\vec{f}) \leq cong_G(\vec{f})$$

Definition

A flow-sparsifier $H$ has quality $\alpha$ if additionally for all demands $\vec{f} \in \mathcal{R}^{(K)}$

$$cong_G(\vec{f}) \leq \alpha cong_H(\vec{f})$$
Definition

A graph $H = (K, E_H)$ on just the terminal set is a **Flow-Sparsifier** if for all demands $\vec{f} \in \mathcal{R}^{(K)}_2$

$$cong_H(\vec{f}) \leq cong_G(\vec{f})$$

Definition

A flow-sparsifier $H$ has quality $\alpha$ if additionally for all demands $\vec{f} \in \mathcal{R}^{(K)}_2$

$$cong_G(\vec{f}) \leq \alpha cong_H(\vec{f})$$

Quality measures how faithfully $H$ approximates $G$ as a communication network.
Our Results I

Theorem

For any undirected, capacitated graph $G = (V, E)$ and any set $K \subset V$ of $k$ terminals, there is an $O(\frac{\log k}{\log \log k})$-quality flow-sparsifier $H = (K, E_H)$. This improves to $O(1)$ if $G$ is planar (or excludes any fixed minor).
Our Results

Theorem

For any undirected, capacitated graph $G = (V, E)$ and any set $K \subset V$ of $k$ terminals, there is an $O(\log k / \log \log k)$-quality flow-sparsifier $H = (K, E_H)$.

This improves to $O(1)$ if $G$ is planar (or excludes any fixed minor).
Our Results I

Theorem

For any undirected, capacitated graph $G = (V, E)$ and any set $K \subset V$ of $k$ terminals, there is an $O(\log k / \log \log k)$-quality flow-sparsifier $H = (K, E_H)$.

This improves to $O(1)$ if $G$ is planar (or excludes any fixed minor)

Theorem

There is a polynomial (in $n$ and $k$) time algorithm to compute a $O(\log^2 k / \log \log k)$-quality flow-sparsifier
[M, 09]: There is a graph $H = (K, E_H)$ so that the cut-function of $H$ approximates minimum cuts in $G$ (separating subsets of terminals)
[M, 09]: There is a graph $H = (K, E_H)$ so that the cut-function of $H$ approximates minimum cuts in $G$ (separating subsets of terminals).

We will refer to this as **Cut Sparsification**.
[M, 09]: There is a graph $H = (K, E_H)$ so that the cut-function of $H$ approximates minimum cuts in $G$ (separating subsets of terminals)

We will refer to this as **Cut Sparsification**

Flow Sparsification is **harder** than Cut Sparsification
Our Results I

Theorem

For any undirected, capacitated graph $G = (V, E)$ and any set $K \subset V$ of $k$ terminals, there is an $O(\log k / \log \log k)$-quality flow-sparsifier $H = (K, E_H)$.

This improves to $O(1)$ if $G$ is planar (or excludes any fixed minor).

Theorem

There is a polynomial (in $n$ and $k$) time algorithm to compute a $O(\log^2 k / \log \log k)$-quality flow-sparsifier.
Our Results II

Theorem

There is an infinite family of graphs and sets of terminals for which any flow-sparsifier has quality $\Omega(\log \log k)$
Theorem

There is an infinite family of graphs and sets of terminals for which any flow-sparsifier has quality $\Omega(\log \log k)$

Motivation for vertex sparsification: obtain approximation algorithms with guarantees independent of $n$
Our Results II

**Theorem**

There is an infinite family of graphs and sets of terminals for which any flow-sparsifier has quality $\Omega(\log \log k)$

**Motivation** for vertex sparsification: obtain approximation algorithms with guarantees independent of $n$

Approximation algorithms that reduce to a $k$-terminal graph must lose a **super-constant** factor in the approximation guarantee
Outline

1. Alternate Definition of Quality
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
3. Lower Bound for Flow Sparsification
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
3. Lower Bound for Flow Sparsification
4. Open Questions
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
3. Lower Bound for Flow Sparsification
4. Open Questions
Suppose we are given a flow-sparsifier $H$. Can we compute the quality of $H$?
Computing Quality

Suppose we are given a flow-sparsifier $H$

**Question**

*Can we compute the quality of $H$?*
Computing Quality
Computing Quality

G

H

a b c d

a b c
d

b d

Computing Quality

\[ f_H = \begin{bmatrix} c_H(a, b) \\ c_H(a, c) \\ c_H(a, d) \\ c_H(b, c) \\ c_H(b, d) \\ c_H(c, d) \end{bmatrix} \]
Computing Quality

\[ \mathbf{f}_H = \begin{bmatrix} c_H(a,b) \\ c_H(a,c) \\ c_H(a,d) \\ c_H(b,c) \\ c_H(b,d) \\ c_H(c,d) \end{bmatrix} \]
Computing Quality

Suppose we are given a flow-sparsifier $H$

**Question**

*Can we compute the quality of $H$?*
Computing Quality

Suppose we are given a flow-sparsifier $H$

**Question**

*Can we compute the quality of $H$?*

The quality is at least $cong_G(\vec{f}_H)$
Computing Quality

G

H

a
b
c
d

a
b
c
d
Computing Quality

\[ \mathbf{f} = \begin{bmatrix} x \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \]
Computing Quality

$$G \Rightarrow H \Rightarrow \begin{bmatrix} x \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
Computing Quality

\[ \mathbf{f} = \begin{bmatrix} x \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \]
Computing Quality

\[ f = \begin{bmatrix} x \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \]
Suppose we are given a flow-sparsifier $H$

**Question**

*Can we compute the quality of $H$?*

The quality is at least $\text{cong}_G(\vec{f}_H)$
Suppose we are given a flow-sparsifier $H$

**Question**

*Can we compute the quality of $H$?*

The quality is at least equal to $cong_G(\vec{f}_H)$
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
3. Lower Bound for Flow Sparsification
4. Open Questions
Outline

1. Alternate Definition of Quality
2. **Geometric Interpretation of Vertex Sparsification**
3. Lower Bound for Flow Sparsification
4. Open Questions
Geometry of Flow Sparsification

\[ \mathcal{P}_G = \{ f \mid \text{cong}_G(f) < 1 \} \]
Geometry of Flow Sparsification

\[ P_G = \{ \vec{f} \mid \text{cong}_G (\vec{f}) < 1 \} \]
Geometry of Flow Sparsification

\[ P_G = \{ f \mid \text{cong}_G(f) < 1 \} \]
$P_G = \{ f \mid \text{cong}_G(f) < 1 \}$
$P_G = \{ \vec{f} \mid \text{cong}_G(\vec{f}) < 1 \}$
Geometry of Flow Sparsification

\[ P_G = \{ \vec{f} \mid \text{cong}_G(f) < 1 \} \]
Geometry of Flow Sparsification

\[ P_G = \{ \vec{f} \mid \text{cong}_G(\vec{f}) < 1 \} \]
What About Cut Sparsification?
What About Cut Sparsification?
What About Cut Sparsification?
What About Cut Sparsification?
What About Cut Sparsification?
What About Cut Sparsification?
What About Cut Sparsification?

[This paper] $\vec{f}_C$

[This paper] $\vec{f}_H$

$\infty P_G$

[M, ’09]
What About Cut Sparsification?

[M, ’09] \( \vec{f}_C \)

[this paper] \( \vec{f}_H \)
What About Cut Sparsification?

[M, ’09] $f_C$

[Max concurrent flow] $f_H$

[this paper]
What About Cut Sparsification?

[M, ’09]

[this paper]
What About Cut Sparsification?

\[ f_{\bar{C}} \quad \text{[M, '09]} \]

\[ f_{\bar{H}} \quad \text{[this paper]} \]

[generalized sparsest cut]
What About Cut Sparsification?
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
3. Lower Bound for Flow Sparsification
4. Open Questions
Outline

1. Alternate Definition of Quality
2. Geometric Interpretation of Vertex Sparsification
3. Lower Bound for Flow Sparsification
4. Open Questions
Proving Lower Bounds on Expanders
Proving Lower Bounds on Expanders
Proving Lower Bounds on Expanders

\[ f_H \]

\[ Q_G \]
The Graph

G'
high-girth, expander

b
c
d
a
The Graph

$G'$

high-girth, expander

$x = \omega(1)$
The Graph

$G'$

high-girth, expander

$x = \omega(1)$
The Graph

$G'$

high-girth, expander

$x = \omega(1)$
The Graph

$G'$

high-girth, expander

$x = \omega(1)$

$G$
The Graph

\[ G' \]
high-girth, expander

\[ G \]

\[ x = \omega(1) \]
The Graph

$G'$

high-girth, expander

$x = \omega(1)$

$G$

$c \ c_1 \ c_2 \ c_3$

$b \ b_1 \ b_2 \ b_3$

$d \ d_1 \ d_2 \ d_3$

$a \ a_1 \ a_2 \ a_3$
The Graph

\( G \)

\( G' \)

high-girth, expander

\( x = \omega(1) \)
A Thought Experiment

$G'$
high-girth, expander

$x = \omega(1)$

$H$

$G$

$x = \omega(1)$
A Thought Experiment

G' high-girth, expander

x = omega(1)

H

??
A Thought Experiment

$G'$

high-girth, expander

$x = \omega(1)$

$H$

??
A Thought Experiment

$G'$

high-girth, expander

$x = \omega(1)$

$H$

$G$

$\omega$
A Thought Experiment

$G'$

high-girth, expander

$x = \omega(1)$

$G$

$H$

??
A Thought Experiment

G'
high-girth, expander
x = \omega(1)

G

H
??
A Thought Experiment

$G'$

high−girth, expander

$x = \omega(1)$

$H$

??

$G$

$\gamma = \omega(1)$

$\gamma'$

??

$\gamma''$

??

$\gamma'''$

??

$\gamma''''$

??
A Thought Experiment

$G'$

high-girth, expander

$x = \omega(1)$

$G$

$H$

??
A Thought Experiment

on average:

\[ d^2 \]

\[ a \quad a_1 \quad a_2 \quad a_3 \]

\[ b \quad b_1 \quad b_2 \quad b_3 \]

\[ c \quad c_1 \quad c_2 \quad c_3 \]

\[ d \quad d_1 \quad d_2 \quad d_3 \]

G'

high-girth, expander

G

\[ a \quad a_1 \quad a_2 \quad a_3 \]

\[ b \quad b_1 \quad b_2 \quad b_3 \]

\[ c \quad c_1 \quad c_2 \quad c_3 \]

\[ d \quad d_1 \quad d_2 \quad d_3 \]

H

x = \text{omega}(1)

on average:

??
A Thought Experiment

G'
high-girth, expander

x = omega(1)

G

on average:
send a unit flow to x terminals
A Thought Experiment

on average:

\[ \text{send a unit flow to } x \text{ terminals} \]
A Thought Experiment

**G’**

high-girth, expander

**G**

**H**

x = \omega(1)

**on average:**

send a unit flow to x terminals

using short paths
A Thought Experiment

on average:
send a unit flow to $x$ terminals
using short paths
A Thought Experiment

Using paths

\[ \begin{align*}
&G' \\
&\text{high-girth, expander} \\
&b \\
&c \\
&d \\
&x = \omega(1)
\end{align*} \]

\[ \begin{align*}
&G \\
&b_1 \\
&b_2 \\
&b_3 \\
&d_1 \\
&d_2 \\
&d_3
\end{align*} \]

\[ \begin{align*}
&H \\
&\text{on average:} \\
&\text{send a unit flow to } x \text{ terminals} \\
&\text{using short paths}
\end{align*} \]
A Thought Experiment

\[ G' \]

high-girth, expander

\[ G \]

\[ x = \omega(1) \]

\[ H \]

on average:

send a unit flow to \( x \) terminals using short paths

not local
A Thought Experiment

G

send a unit flow to x terminals
on average:
short using          paths
local reaching only          terminals

x = omega(1)
high−girth, expander
H

on average:
send a unit flow to x terminals
using short paths
reaching only local terminals

not local
Girth-Routed Edges
Girth-Routed Edges
Girth-Routed Edges

G

H
Girth-Routed Edges
Girth-Routed Edges

not too many edges can be routed on girth paths
Girth-Routed Edges

not too many edges can be routed on girth paths
A Reduction to Cut-Width
A Reduction to Cut-Width
A Reduction to Cut-Width
A Reduction to Cut-Width
A Reduction to Cut-Width
A Reduction to Cut-Width
A Reduction to Cut-Width
A Reduction to Cut-Width
Recent work on both upper bounds and lower bounds
Recent Work

Recent work on both upper bounds and lower bounds

1. [Englert, Gupta, Krauthgamer, Raecke, Talgam, Talwar]
2. [Makarychev, Makarychev]
3. [Charikar, Leighton, Li, M]
Recent work on both upper bounds and lower bounds

1. [Englert, Gupta, Krauthgamer, Raecke, Talgam, Talwar]
2. [Makarychev, Makarychev]
3. [Charikar, Leighton, Li, M]

... super-constant lower bounds for cut sparsification, and constructive results that match our existential results
Question

*Can the approximation for 0-extension be improved?*
Open Questions

Question

*Can the approximation for 0-extension be improved?*

Question

*What if the semi-metric is $\ell_1$?*

(immediately implies improvements to cut-sparsification)
Open Questions

Question

*Can the approximation for 0-extension be improved?*

Question

*What if the semi-metric is $\ell_1$?*

(immediately implies improvements to cut-sparsification)

Question

*Is flow-sparsification easier than the 0-extension problem?*
Questions?
Thanks!