Flow-Based Algorithms for Local Graph Clustering

Lorenzo Orecchia (MIT Math)  Zeyuan A. Zhu (MIT CSAIL)
Graph Clustering for Large Networks

**INPUT:** Large Data or Social Network

**GOAL:** find clusters, i.e., subsets well-connected inside and poorly connected to the rest of the graph.
INPUT: Large Data or Social Network

GOAL: find clusters, i.e.,
subsets well-connected inside and poorly connected to the rest of the graph.
Graph Clustering and Conductance

Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Measure of Cluster Quality is Conductance:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{Vol}(S), \text{Vol}(\bar{S})\}}$$
Graph Clustering and Conductance

Undirected unweighted $G = (V, E)$, $|V| = n$, $|E| = m$

Measure of Cluster Quality is Conductance:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

$$\text{vol}(S) = |E(S, V)| = 8$$

$$|E(S, \bar{S})| = 4$$

$$\phi(S) = \frac{1}{2}$$
Graph Clustering and Conductance

Undirected unweighted $G = (V, E)$, $|V| = n$, $|E| = m$

Measure of Cluster Quality is Conductance:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

Random-Walk Interpretation

Given uniform distribution over $S$, what is probability of exiting $S$ in one step of random walk over the edges of $G$?
Graph Clustering and Conductance

Undirected unweighted \( G = (V, E), |V| = n, |E| = m \)

Measure of Cluster Quality is Conductance:

\[
\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}
\]

Fundamental primitive in graph clustering: find cut of minimum conductance

\[
\min_{S \subseteq V} \phi(S)
\]
Graph Clustering and Conductance

Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Measure of Cluster Quality is Conductance:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

Fundamental primitive in graph clustering: find cut of minimum conductance

$$\min_{S \subseteq V} \phi(S)$$
Graph Clustering and Conductance

Undirected unweighted $G = (V, E)$, $|V| = n$, $|E| = m$

Measure of Cluster Quality is Conductance:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}({\bar{S}})\}}$$

Fundamental primitive in graph clustering: find cut of minimum conductance

$$\min_{S \subseteq V} \phi(S)$$
Graph Clustering and Conductance

Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Measure of Cluster Quality is Conductance:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

Fundamental primitive in graph clustering: find cut of minimum conductance

$$\min_{S \subseteq V} \phi(S)$$
Graph Clustering and Conductance

Undirected unweighted \( G = (V, E), |V| = n, |E| = m \)

Measure of Cluster Quality is Conductance:

\[
\phi(S) = \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}
\]

Fundamental primitive in graph clustering: find cut of minimum conductance

\[
\min_{S \subseteq V} \phi(S)
\]

- Fundamental NP-Complete problem
- Best known approximation algorithm achieves \( O(\sqrt{\log n}) \)-approximation via SDP relaxation
- SDP relaxation can be interpreted as combining spectral and flow techniques
Local Clustering for Large Networks

\[ G = (V, E), \quad |V| = n, |E| = m \]

Massive Networks

\[ n \to \infty, \quad m \to \infty \]
Local Clustering for Large Networks

\[ G = (V, E), \quad |V| = n, |E| = m \]

Massive Networks

\[ n \to \infty, \ m \to \infty \]
Local Clustering for Large Networks

 Massive Networks

$G = (V, E)$,
$|V| = n, |E| = m$

$n \to \infty, m \to \infty$

 Semi-supervised Model

$\min_{S \subseteq V} \phi(S)$

Detect low-conductance cut “near” input region $A$
Local Graph Clustering Problem

\[ \min_{S \subseteq V} \phi(S) \]

Semi-supervised Model

Detect low-conductance cut “near” input region \( A \)

Problem formulation:
Given random vertex from target cut \( A \), find cut \( S \) such that

\[ \phi(S) \approx \phi(A) \]

in time \( \text{poly}(\text{Vol}(A)) \).

Infinite graph \( G \)

Random seed

Target cut \( A \)
Local Graph Clustering Problem

\[ \min_{S \subseteq V} \phi(S') \]

Semi-supervised Model

Detect low-conductance cut “near” input region \( A \)

Problem formulation:
Given random vertex from target cut \( A \), find cut \( S \) such that

\[ \phi(S') \approx \phi(A) \]

in time \( \text{poly}(|\text{Vol}(A)|) \). In particular, explored region has size \( \text{poly}(|\text{Vol}(A)|) \).

Infinite graph \( G \)

Explored region

Output Cut \( S \)
Random-Walk-Based Algorithms

Compute marginals of random walk started at seed
Random-Walk-Based Algorithms

Compute marginals of random walk started at seed

Vector of marginals

High probability

Low probability
Random-Walk-Based Algorithms

Compute marginals of random walk started at seed

High probability

Low probability

Output sweep cut of minimum conductance
Random-Walk-Based Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Output Conductance</th>
<th>Support Size</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Spielman, Teng’04]</td>
<td>$\sqrt{\phi(A) \log^3 \text{vol}(A)}$</td>
<td>$O(\text{Vol}(A))$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\phi(A)^2}\right)$</td>
</tr>
<tr>
<td>[Andersen, Chung, Lang ‘07]</td>
<td>$\sqrt{\phi(A) \log \text{vol}(A)}$</td>
<td>$O(\text{Vol}(A))$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\phi(A)}\right)$</td>
</tr>
<tr>
<td>[Andersen, Peres ‘09]</td>
<td>$\sqrt{\phi(A) \log \text{vol}(A)}$</td>
<td>$O(\text{Vol}(A))$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\sqrt{\phi(A)}}\right)$</td>
</tr>
<tr>
<td>[Oveis Gharan, Trevisan ’12]</td>
<td>$\sqrt{\frac{\phi(A)}{\epsilon}}$</td>
<td>$O(\text{Vol}(A)^{1+\epsilon})$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\sqrt{\phi(A)}}\right)$</td>
</tr>
</tbody>
</table>
# Random-Walk-Based Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Output Conductance</th>
<th>Support Size</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Spielman, Teng’04]</td>
<td>$\sqrt{\phi(A) \log^3 \text{vol}(A)}$</td>
<td>$O(\text{Vol}(A))$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\phi(A)^2}\right)$</td>
</tr>
<tr>
<td>[Andersen, Chung, Lang ‘07]</td>
<td>$\sqrt{\phi(A) \log \text{vol}(A)}$</td>
<td>$O(\text{Vol}(A))$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\phi(A)}\right)$</td>
</tr>
<tr>
<td>[Andersen, Peres ‘09]</td>
<td>$\sqrt{\phi(A) \log \text{vol}(A)}$</td>
<td>$O(\text{Vol}(A))$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\sqrt{\phi(A)}}\right)$</td>
</tr>
<tr>
<td>[Oveis Gharan, Trevisan ‘12]</td>
<td>$\sqrt{\frac{\phi(A)}{\epsilon}}$</td>
<td>$O(\text{Vol}(A)^{1+\epsilon})$</td>
<td>$\tilde{O}\left(\frac{\text{Vol}(A)}{\sqrt{\phi(A)}}\right)$</td>
</tr>
</tbody>
</table>

Essentially optimal for Random-Walk approach
**Well-Connected Local Clustering**

**OBSERVATION:**  
Spectral, random-walk-based algorithms in practice work better than theoretical bounds.

**RECENT WORK** (both in global and local graph clustering):  
Strengthen assumptions to explain behavior of spectral algorithms:

**NEW ASSUMPTION:** “A good ground-truth clustering exists”

**GLOBAL SETTING:** bounds on higher eigenvalues.

**LOCAL SETTING:** mixing time $\tau_A$ of graph induced by target cut $A$. 
Well-Connected Local Clustering

**OBSERVATION:**
Spectral, random-walk-based algorithms *in practice work better than theoretical bounds*

**RECENT WORK** *(both in global and local graph clustering):*
Strengthen assumptions to explain behavior of spectral algorithms:

**NEW ASSUMPTION:** “A good ground-truth clustering exists”

**GLOBAL SETTING:** bounds on higher eigenvalues.

**LOCAL SETTING:** mixing time $\tau_A$ of graph induced by target cut $A$.

Input cluster $A$ should be well connected
Well-Connected Local Clustering

**OBSERVATION:** Spectral, random-walk-based algorithms in practice work better than theoretical bounds

**RECENT WORK** (both in global and local graph clustering):
Strengthen assumptions to explain behavior of spectral algorithms:

**NEW ASSUMPTION:** “A good ground-truth clustering exists”

**GLOBAL SETTING:** bounds on higher eigenvalues.

**LOCAL SETTING:** mixing time $\tau_A$ of graph induced by target cut $A$.

Input cluster $A$ should be well connected
Well-Connected Local Clustering

**OBSERVATION:**
Spectral, random-walk-based algorithms *in practice work better than theoretical bounds*

**RECENT WORK** *(both in global and local graph clustering):*
Strengthen assumptions to explain behavior of spectral algorithms:

**NEW ASSUMPTION:** "A good ground-truth clustering exists"

**LOCAL SETTING:** mixing time $\tau_A$ of graph induced by target cut $A$.

[Zhu, Lattanzi, Mirrokni ’13] Under the well-connectedness assumption

$$\tau_A \cdot O \left( \frac{1}{\phi(A)} \right)$$

The random-walk-based algorithm outputs cut $S$ with

$$\phi(S) \cdot O(\sqrt{\tau_A}) \cdot \phi(A)$$

**MIXING TIME WITHIN SET $A$ SMALLER THAN TO THE OUTSIDE**

and

$$\text{vol}(\text{Explored}) \cdot O(\text{vol}(A))$$

**APPROXIMATION RATIO**

**LOCALITY GUARANTEE**
Limitations of Random-Walks

[Zhu, Lattanzi, Mirrokni ’13] Under the well-connectedness assumption

\[ \tau_A \cdot O \left( \frac{1}{\phi(A)} \right) \]

The random-walk-based algorithm outputs cut \( S \) with

\[ \phi(S) \cdot O(\sqrt{\tau_A}) \cdot \phi(A) \quad \text{and} \quad \text{vol(Explored}) \cdot O(\text{vol}(A)) \]

LIMITATIONS:

- This result is also essentially tight for the random-walk approach
- Only weak pseudo-approximation, when \( \phi(A) \) is small
Limitations of Random-Walks

[Zhu, Lattanzi, Mirrokni ’13] Under the well-connectedness assumption

$$\tau_A \cdot O\left(\frac{1}{\phi(A)}\right)$$

The random-walk-based algorithm outputs cut $S$ with

$$\phi(S') \cdot O(\sqrt{\tau_A}) \cdot \phi(A) \quad \text{and} \quad \text{vol(Explored))} \cdot O(\text{vol}(A))$$

**APPROXIMATION RATIO**

**LOCALITY GUARANTEE**

**LIMITATIONS:**

- This result is also essentially tight for the random-walk approach

- Only weak pseudo-approximation, when $\phi(A)$ is small

$$\tau_A = \Theta\left(\frac{1}{\phi(A)}\right) \quad \phi(S') \cdot O(\sqrt{\phi(A)})$$

Typical Cheeger-like spectral guarantee
Our Result

OUR RESULT [Orecchia, Zhu’14]:
Under the same well-connectedness assumption,

\[ \tau_A \cdot O\left(\frac{1}{\phi(A)}\right) \]

our algorithm runs in time \( \tilde{O}\left(\frac{\text{Vol}(A)}{\phi(A)}\right) \) and yields

\[ \phi(S) \cdot O(\phi(A)) \quad \text{and} \quad \text{vol(Explored))} \cdot \tilde{O}(\text{vol}(A)) \]

MIXING TIME WITHIN SET \( A \) SMALLER THAN TO THE OUTSIDE
Our Result

OUR RESULT [Orecchia, Zhu’14]:
Under the same well-connectedness assumption,

\[ \tau_A \cdot O\left(\frac{1}{\phi(A)}\right) \]

our algorithm runs in time \( \tilde{O}\left(\frac{\text{Vol}(A)}{\phi(A)}\right) \) and yields

\[ \phi(S) \cdot O\left(\phi(A)\right) \]
and

\[ \text{vol(Explored)} \cdot O(\text{vol}(A)) \]

CONSTANT APPROXIMATION

CONSTANT SIZE BLOW-UP

ALGORITHMIC IDEA:
Post-processes output of random-walk algorithm by localized flow computation
Overlap Property

ADDITIONAL PROPERTY OF RANDOM-WALK–BASED ALGORITHM:

[Zhu, Lattanzi, Mirrokni ’13] Under the well-connectedness assumption

\[ \tau_A \cdot O \left( \frac{1}{\phi(A)} \right) \]

MIXING TIME WITHIN SET \( A \) SMALLER THAN TO THE OUTSIDE

The random-walk-based algorithm outputs cut \( S \) with

\[ \frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \Box(1). \]

OVERLAP PROPERTY
Overlap Property

ADDITIONAL PROPERTY OF RANDOM-WALK–BASED ALGORITHM:

[Zhu, Lattanzi, Mirrokni ’13] Under the well-connectedness assumption

\[ \tau_A \cdot O\left(\frac{1}{\phi(A)}\right) \]

The random-walk-based algorithm outputs cut \( S \) with

\[ \frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \square(1). \]

\( \tau_A \) · \( O\left(\frac{1}{\phi(A)}\right) \)

OVERLAP PROPERTY

MIXING TIME WITHIN SET \( A \) SMALLER THAN TO THE OUTSIDE

Output Cut \( S \)  

Unknown Target cut \( A \)

Intersection is constant fraction of volume of target cut
Overlap Property

ADDITIONAL PROPERTY OF RANDOM-WALK –BASED ALGORITHM:

[Zhu, Lattanzi, Mirrokni ’13] Under the well-connectedness assumption

$$\tau_A \cdot O\left(\frac{1}{\phi(A)}\right)$$

The random-walk-based algorithm outputs cut $S$ with

$$\frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \square(1).$$

GOAL: Exploit overlap to produce better approximation to $\phi(A)$
Cut-Improvement Algorithms

[Lang, Rao ‘93][Andersen, Lang ‘08]
Cut-Improvement Algorithms

[Lang, Rao ‘93][Andersen, Lang ‘08]

On input cut $S$, this algorithm outputs cut $C$ such that, for any target cut $A$ with
\[
\frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \frac{\text{vol}(S)}{\text{vol}(V)} + \delta \frac{\text{vol}(\bar{S})}{\text{vol}(V)}
\]

we have
\[
\phi(C) \cdot \frac{1}{\delta} \phi(A)
\]

The algorithm runs a small number of global s-t maxflow computations.
Cut-Improvement Algorithms

[Lang, Rao ‘93][Andersen, Lang ‘08]
On input cut $S$, this algorithm outputs cut $C$ such that, for any target cut $A$ with

$$\frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \frac{\text{vol}(S)}{\text{vol}(V)} + \delta \frac{\text{vol}(\overline{S})}{\text{vol}(V)}$$

we have

$$\phi(C) \cdot \frac{1}{\delta} \phi(A)$$

The algorithm runs a small number of global s-t maxflow computations.
Cut-Improvement Algorithms

[Lang, Rao ‘93][Andersen, Lang ‘08]
On input cut $S$, this algorithm outputs cut $C$ such that, for any target cut $A$ with

\[
\frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \frac{\text{vol}(S)}{\text{vol}(V)} + \delta \frac{\text{vol}(\bar{S})}{\text{vol}(V)} \geq \delta \geq \Box (1)
\]

we have

\[
\phi(C) \cdot \frac{1}{\delta} \phi(A) \cdot O(\phi(A))
\]

The algorithm runs a small number of global s-t maxflow computations.

\[\text{IN OUR CASE: } \text{vol}(V), \text{vol}(\bar{S}) \to \infty\]
Cut-Improvement Algorithms

[Lang, Rao ‘93][Andersen, Lang ‘08]

On input cut $S$, this algorithm outputs cut $C$ such that, for any target cut $A$ with

\[
\frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \frac{\text{vol}(S)}{\text{vol}(V)} + \delta \frac{\text{vol}(\tilde{S})}{\text{vol}(V)} \geq \delta \geq \Box(1)
\]

we have

\[
\phi(C) \cdot \frac{1}{\delta} \phi(A) \cdot O(\phi(A))
\]

The algorithm runs a small number of global s-t maxflow computations.

**IN OUR CASE:**

\[
\text{vol}(V), \text{vol}(\tilde{S}) \rightarrow \infty
\]
Cut-Improvement Algorithms

[Lang, Rao ‘93][Andersen, Lang ‘08]

On input cut $S$, this algorithm outputs cut $C$ such that, for any target cut $A$ with

$$\frac{\text{vol}(S \cap A)}{\text{vol}(A)} \geq \frac{\text{vol}(S)}{\text{vol}(V)} + \delta \frac{\text{vol}(\tilde{S})}{\text{vol}(V)} \geq \delta \geq \square(1)$$

we have

$$\phi(C) \cdot \frac{1}{\delta} \phi(A) \cdot O(\phi(A))$$

The algorithm runs a small number of global s-t maxflow computations.

IN OUR CASE:

$$\text{vol}(V), \text{vol}(\tilde{S}) \rightarrow \infty$$
Global Solution to Cut-Improvement

Input Cut $\mathcal{S}$

Graph $\mathcal{G}$

$\mathcal{S}$
Global Solution to Cut-Improvement

Graph $G$

Source node

Input Cut $\mathcal{S}$

Sink node

$\bar{\mathcal{S}}$
Global Solution to Cut-Improvement

Input Cut $\mathcal{S}$

Source node

Sink node

Graph $G$

Source node

Sink node

$\alpha \cdot d(v)$

$\alpha \cdot \frac{\text{vol}(S)}{\text{vol}(\overline{S})} \cdot d(u)$
Global Solution to Cut-Improvement

Graph $G$

Input Cut $S$

Source node

Sink node

Cut capacity = $\alpha \cdot \text{vol}(S)$

Cut capacity = $\alpha \cdot \frac{\text{vol}(S)}{\text{vol}(\overline{S})} \cdot d(u)$
Global Solution to Cut-Improvement

Cut capacity = $\alpha \cdot \text{vol}(S)$

Solve parametric flow problem:
\[
\max \alpha \\
\text{s.t. total source/sink capacity can be routed}
\]
Global Solution to Cut-Improvement

Input Cut $\mathcal{S}$

Cut capacity = $\alpha \cdot \text{vol}(\mathcal{S})$

PROBLEM: optimal (and nearly-optimal) flow solutions are inherently global
Localization of Cut-Improvement

QUESTION: Can the cut-improvement algorithm be localized?

PESSIMISTIC CONJECTURE: NO.
Localization is an exclusive feature of random-walk-based algorithms.
Localization of Cut-Improvement

QUESTION: Can the cut-improvement algorithm be localized?

PESSIMISTIC CONJECTURE: NO.
Localization is an exclusive feature of random-walk-based algorithms.

NB: Global computation also required for exact random-walk marginals.
Localization of Cut-Improvement

**QUESTION**: Can the cut-improvement algorithm be localized?

**PESSIMISTIC CONJECTURE**: NO.

Localization is an exclusive feature of random-walk-based algorithms.

**NB**: Global computation also required for exact random-walk marginals.

**HOW DOES RANDOM WALK APPROACH ACHIEVE LOCALIZATION?**
Localization of Cut-Improvement

**QUESTION**: Can the cut-improvement algorithm be localized?

**PESSIMISTIC CONJECTURE**: NO. Localization is an exclusive feature of random-walk-based algorithms.
Localization of Random-Walk

Marginals are computed by "pushing" probability mass from vertices:

Argument applies to PageRank random walk, but can be generalized to other walks.
Localization of Random-Walk

Marginals are computed by "pushing" probability mass from vertices:

Argument applies to PageRank random walk, but can be generalized to other walks.
Localization of Random-Walk

Marginals are computed by "pushing" probability mass from vertices:

Argument applies to PageRank random walk, but can be generalized to other walks.
Localization of Random-Walk

Marginals are computed by "pushing" probability mass from vertices:

Localization is achieved by pushing mass only if current mass at vertex is larger than a certain threshold.
Modification of Flow Problem Achieves Localization

Input Cut $S$

$\alpha \cdot d(v)$

Source node

$\alpha \cdot \epsilon \cdot d(u)$

Sink node

Graph $G$

$\bar{S}$

Cut capacity = $\alpha \cdot \text{vol}(S)$

Every sink-side node has a large capacity to the sink: $\epsilon = \Box(1)$
Modification of Flow Problem Achieves Localization

Cut capacity = $\alpha \cdot \text{vol}(S)$

$\epsilon = \square(1)$

Cut capacity = $\alpha \cdot \epsilon \cdot \text{vol}(\bar{S})$

**CONSEQUENCE:** There exist local optimal solutions:

$$\text{vol(Explored)} \cdot \frac{\alpha \text{vol}(S)}{\alpha \epsilon} = \frac{\text{vol}(S)}{\epsilon}$$
Modification of Flow Problem Achieves Localization

**Input Cut**

\[ S \]

\[ \text{Cut capacity} = \alpha \cdot \text{vol}(S) \]

\[ \epsilon = \Box(1) \]

\[ \text{Cut capacity} = \alpha \cdot \epsilon \cdot \text{vol}(\overline{S}) \]

**CONSEQUENCE:** There exist local optimal solutions: \( \text{vol(Explored)} \cdot \frac{\text{vol}(S)}{\epsilon} \)

**LAST OBSTACLE:** Is this modification still able to solve cut-improvement problem?
Optimization Interpretation

GLOBAL CUT-IMPROVEMENT on input $S$:

$$\min_{C \subseteq V} \frac{\phi(C) \left( \frac{\text{vol}(C \cap S)}{\text{vol}(S)} - \frac{\text{vol}(C \cap \bar{S})}{\text{vol}(\bar{S})} \right)}{\text{vol}(C \setminus S)}$$

**NB:** can be solved by maxflow thanks to maxflow-mincut theorem.
GLOBAL CUT-IMPROVEMENT on input $S$: 

\[ \min_{C \subseteq V} \frac{\phi(C)}{\frac{\text{vol}(C \cap S)}{\text{vol}(S)} - \frac{\text{vol}(C \cap \bar{S})}{\text{vol}(\bar{S})}} \]

**NB:** can be solved by maxflow thanks to maxflow-mincut theorem.
Optimization Interpretation

GLOBAL CUT-IMPROVEMENT on input $S$:

$$\min_{C \subseteq V} \frac{\phi(C)}{\frac{\text{vol}(C \cap S)}{\text{vol}(S)} - \frac{\text{vol}(C \cap \bar{S})}{\text{vol}(\bar{S})}}$$

**NB:** can be solved by maxflow thanks to maxflow-mincut theorem.

LOCALIZED CUT-IMPROVEMENT – RESTRICT FEASIBLE SETS $C$:

$$\min_{C \subseteq V} \frac{\phi(C)}{\frac{\text{vol}(C \cap S)}{\text{vol}(S)} - \frac{\text{vol}(C \cap \bar{S})}{\text{vol}(\bar{S})}}$$

s.t. $\frac{\text{vol}(C \cap S')}{\text{vol}(C)} \geq \Box(\epsilon)$.  

OVERLAP LOWER BOUND
Optimization Interpretation

LOCALIZED CUT-IMPROVEMENT – RESTRICT FEASIBLE SETS $\mathcal{C}$:

$$\min_{C \subseteq V} \frac{\phi(C')}{\frac{\text{vol}(C \cap S)}{\text{vol}(S)} - \frac{\text{vol}(C \cap \bar{S})}{\text{vol}(\bar{S})}}$$

s.t. \( \frac{\text{vol}(C \cap S)}{\text{vol}(C')} \geq \square(\epsilon). \)

RESULT:
This problem is solved exactly by the localized parametric flow problem.
Future Directions

• Combine local flow and random-walk algorithms in stronger ways

**OPEN QUESTION:** Does there exist a local version of cut-matching game?

It would yield **unconditional polylog approximation** to local graph clustering.

• Exploit optimization interpretation to design more local algorithms.
Future Directions

• Combine local flow and random-walk algorithms in stronger ways

**OPEN QUESTION:** Does there exist a local version of cut-matching game?

It would yield unconditional polylog approximation to local graph clustering.

• Exploit optimization interpretation to design more local algorithms.

THE END – THANK YOU!