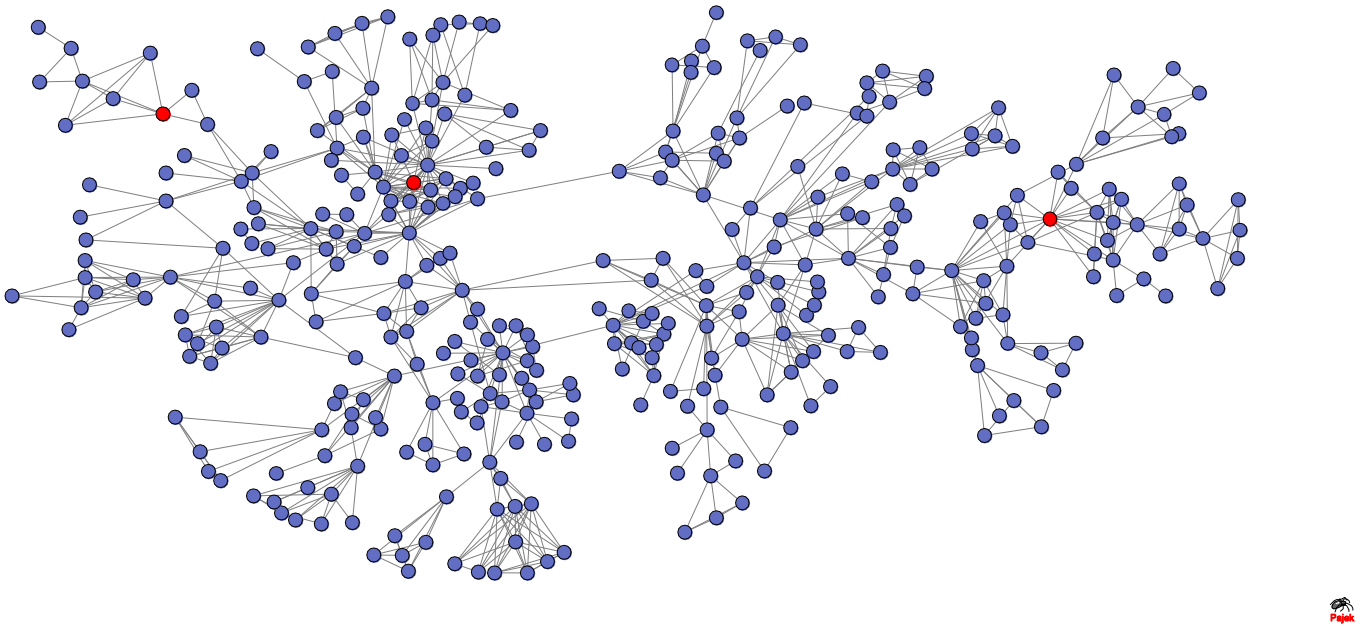
Flow-Based Algorithms for Local Graph Clustering

Lorenzo Orecchia
(MIT Math)

Zeyuan A. Zhu
(MIT CSAIL)

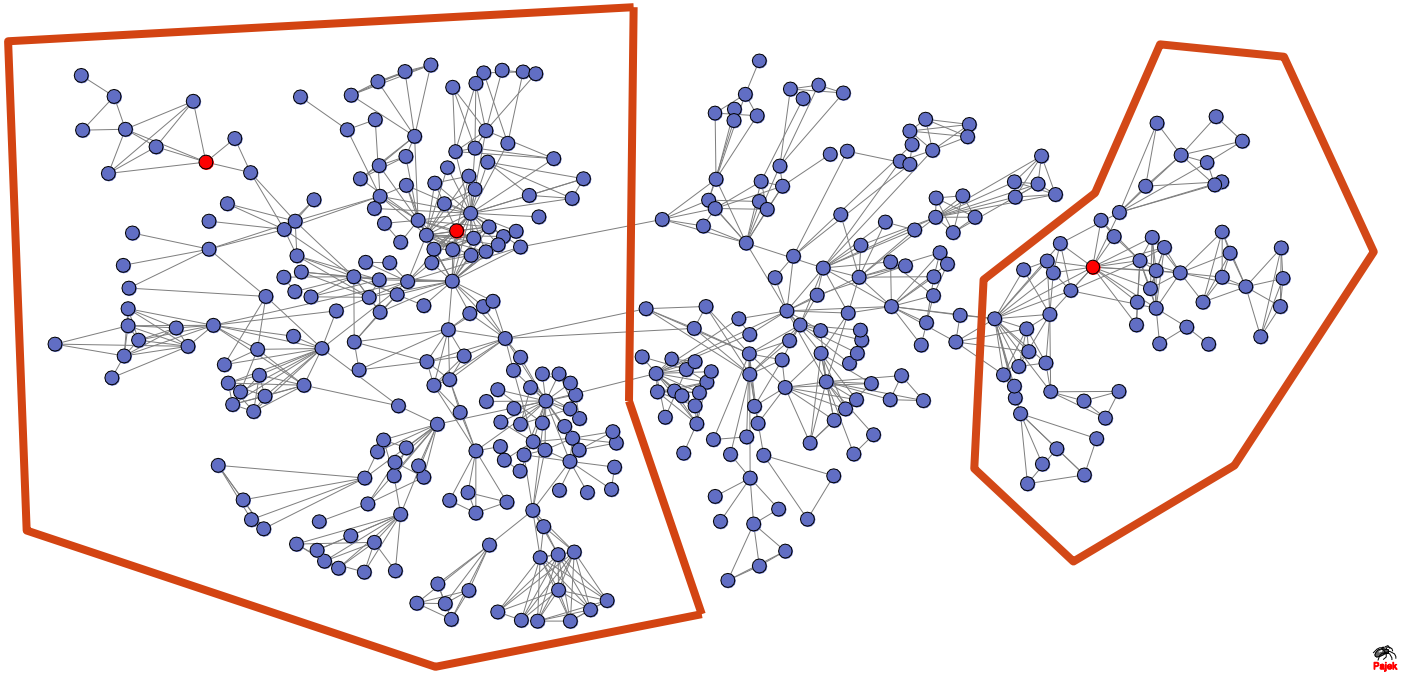
Graph Clustering for Large Networks



INPUT: Large Data or Social Network

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

Graph Clustering for Large Networks

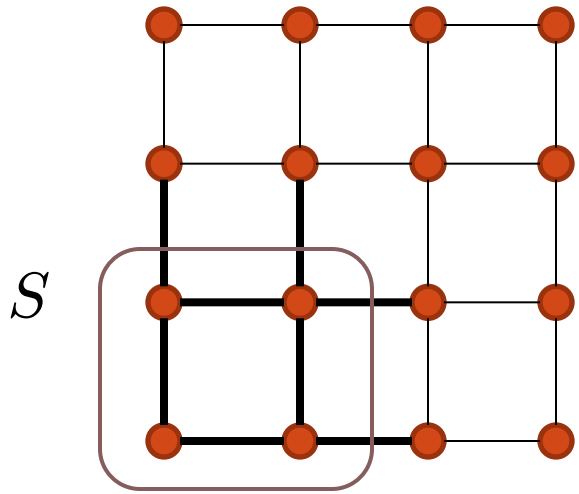


INPUT: Large Data or Social Network

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

Graph Clustering and Conductance

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

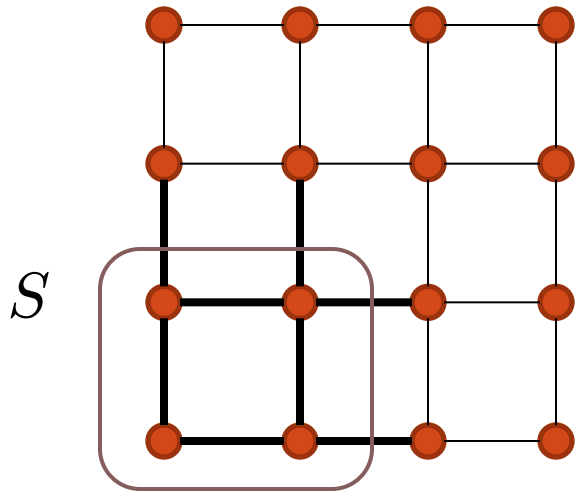


Measure of Cluster Quality is Conductance:

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

Graph Clustering and Conductance

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



Measure of Cluster Quality is Conductance:

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

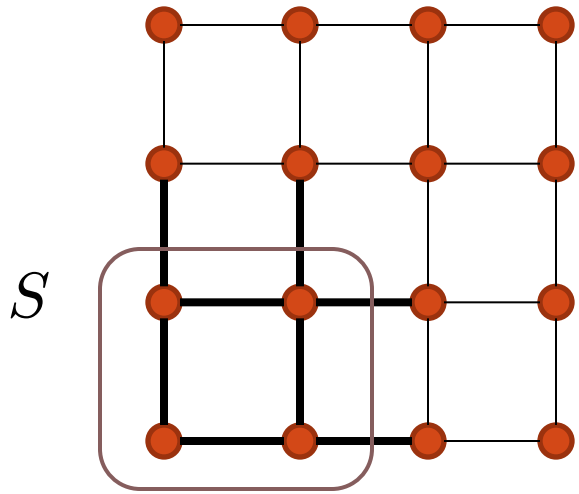
$$\mathrm{vol}(S) = |E(S, V)| = 8$$

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

$$\phi(S) = \frac{1}{2}$$

Graph Clustering and Conductance

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



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

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

$\phi(S) = \frac{1}{2}$

Measure of Cluster Quality is Conductance:

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

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:

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

Fundamental primitive in graph clustering: find cut of minimum conductance

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

Graph Clustering and Conductance

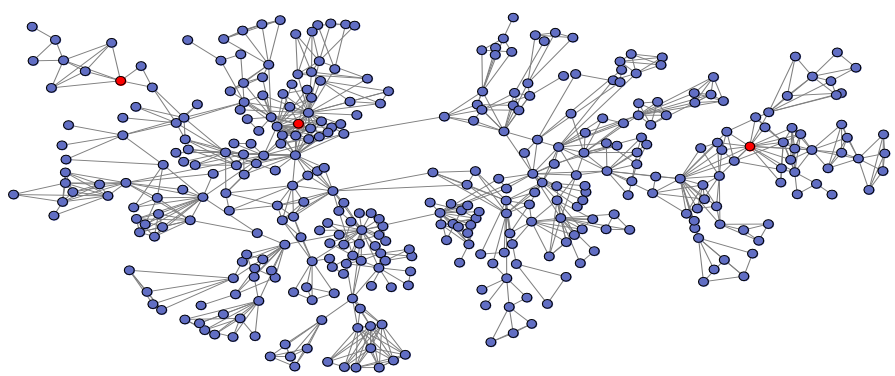
Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Measure of Cluster Quality is Conductance:

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

Fundamental primitive in graph clustering: find cut of minimum conductance

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



Graph Clustering and Conductance

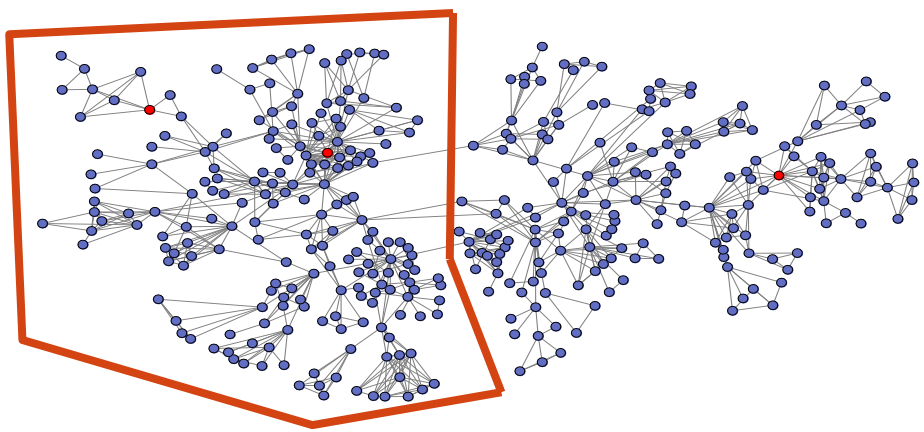
Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Measure of Cluster Quality is Conductance:

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

Fundamental primitive in graph clustering: find cut of minimum conductance

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



Graph Clustering and Conductance

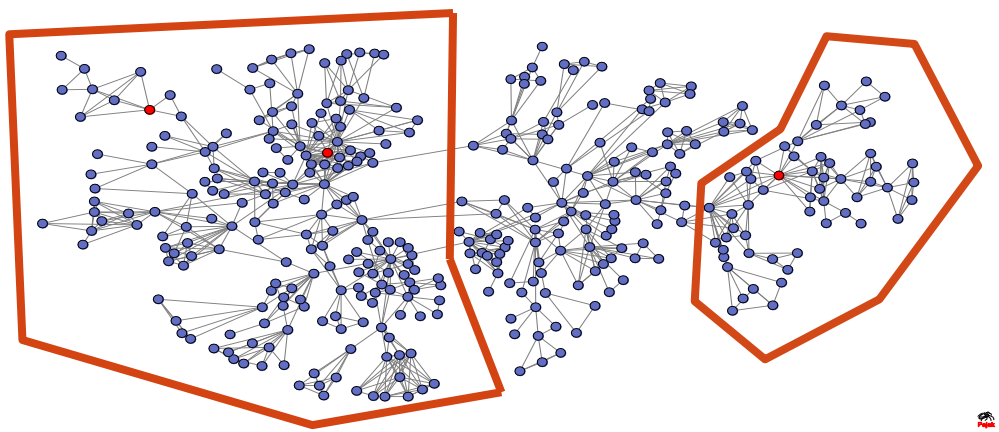
Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Measure of Cluster Quality is Conductance:

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

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:

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

Fundamental primitive in graph clustering: find cut of minimum conductance

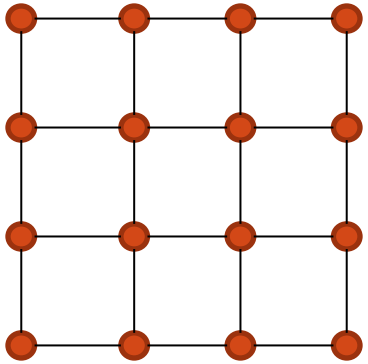
$$\min_{S \subseteq V} \phi(S)$$

- Fundamental NP-Complete problem
- Best known approximation algorithm achieves $O(\sqrt{\log n})$-approximation via SDP relaxation
- SDP relaxation can be interpreted as <u>combining spectral and flow techniques</u>

Local Clustering for Large Networks

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

Massive Networks $\longrightarrow$ $n \to \infty,\ m \to \infty$

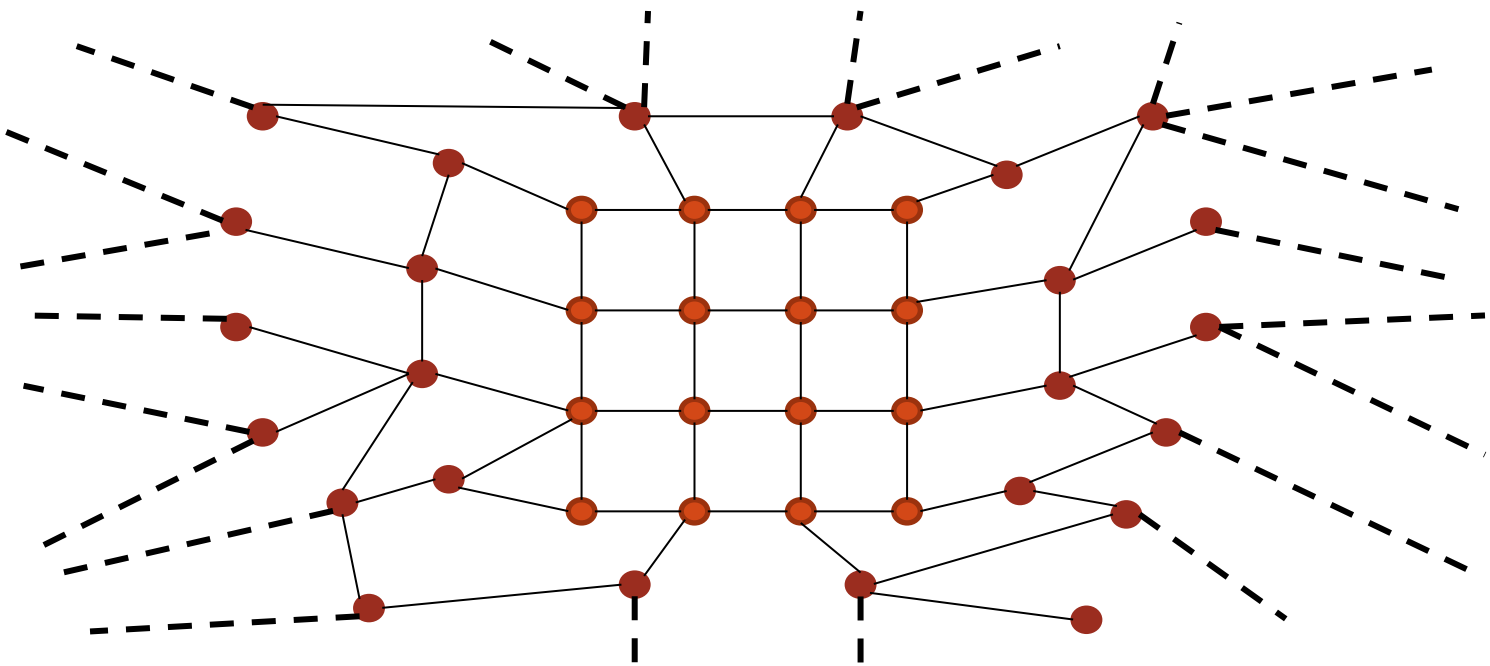


Local Clustering for Large Networks

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

Massive Networks →

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



Local Clustering for Large Networks

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

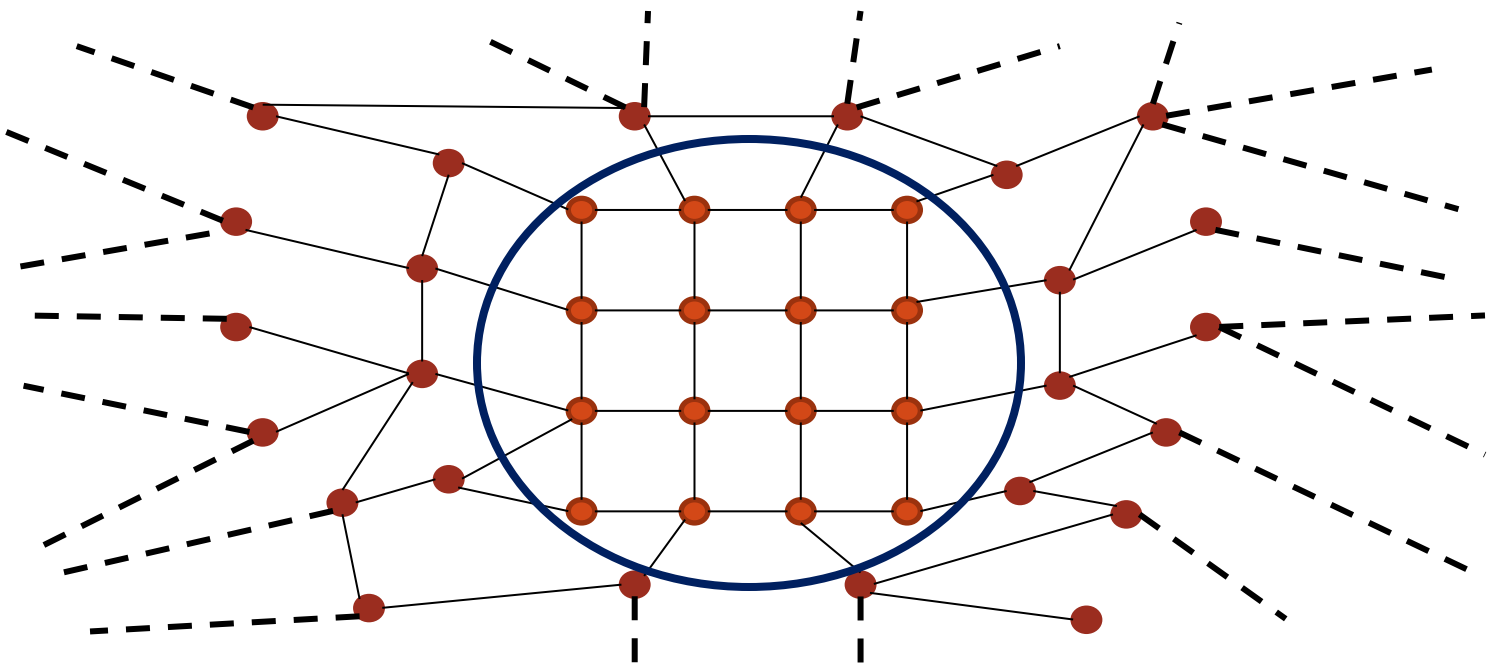
Massive Networks →

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

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

Semi-supervised Model →

Detect low-conductance cut "**near" input region** A



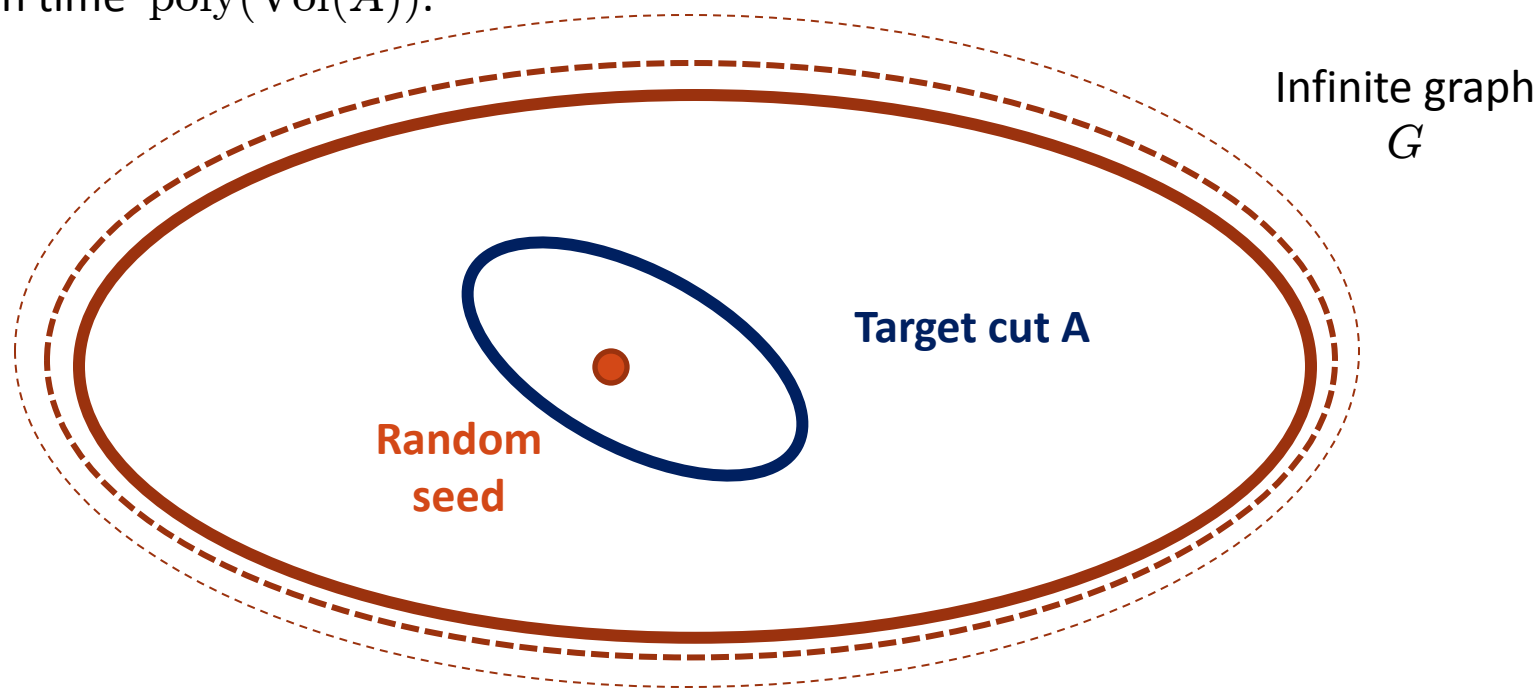
Local Graph Clustering Problem

$\min_{S \subseteq V} \phi(S)$ **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 $\mathrm{poly}(\mathrm{Vol}(A))$.



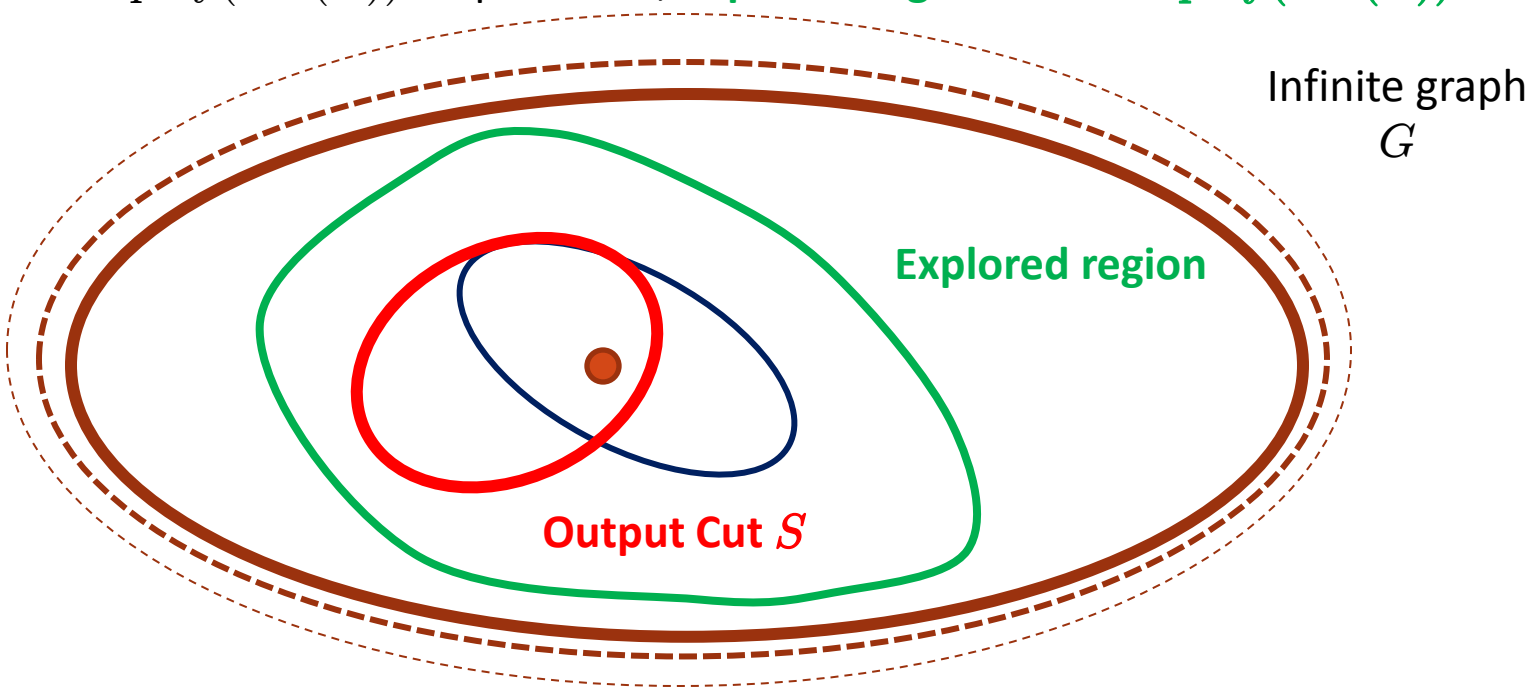
Local Graph Clustering Problem

$\min_{S \subseteq V} \phi(S)$ **Semi-supervised Model** $\longrightarrow$ Detect low-conductance cut "**near" input region A**

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

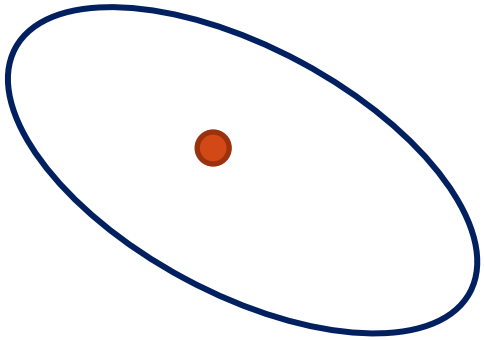
$$\phi(S) \approx \phi(A)$$

in time $\mathrm{poly}(\mathrm{Vol}(A))$. In particular, **explored region has size** $\mathrm{poly}(\mathrm{Vol}(A))$.



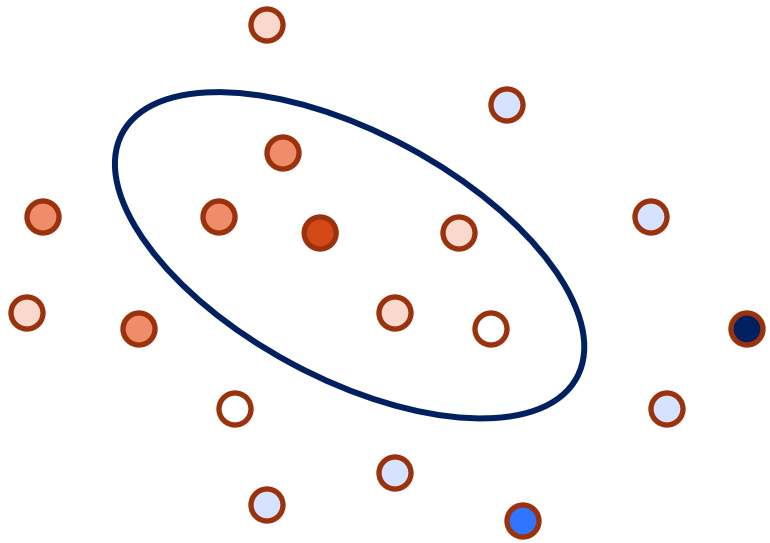
Random-Walk-Based Algorithms

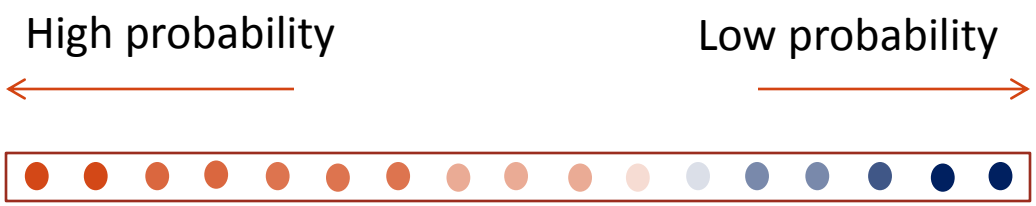
Compute marginals of **random walk started at seed**



Random-Walk-Based Algorithms

Compute marginals of **random walk started at seed**

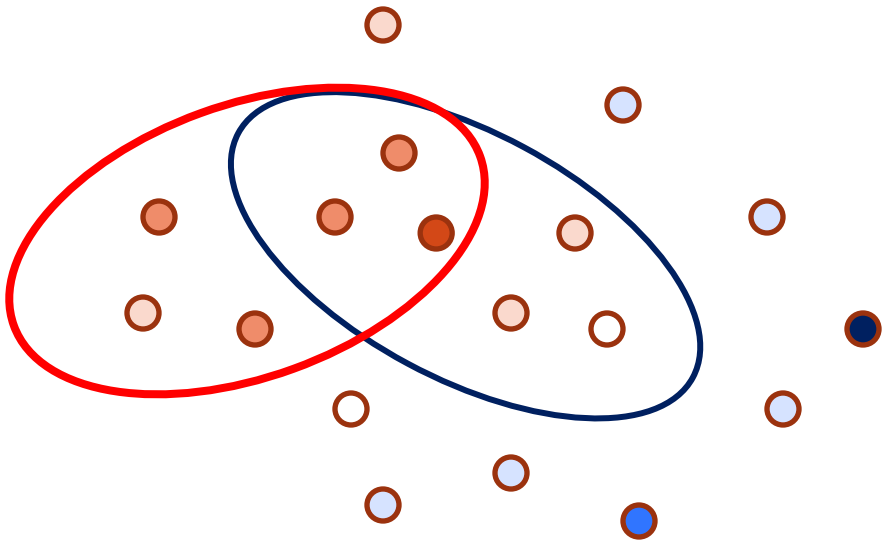


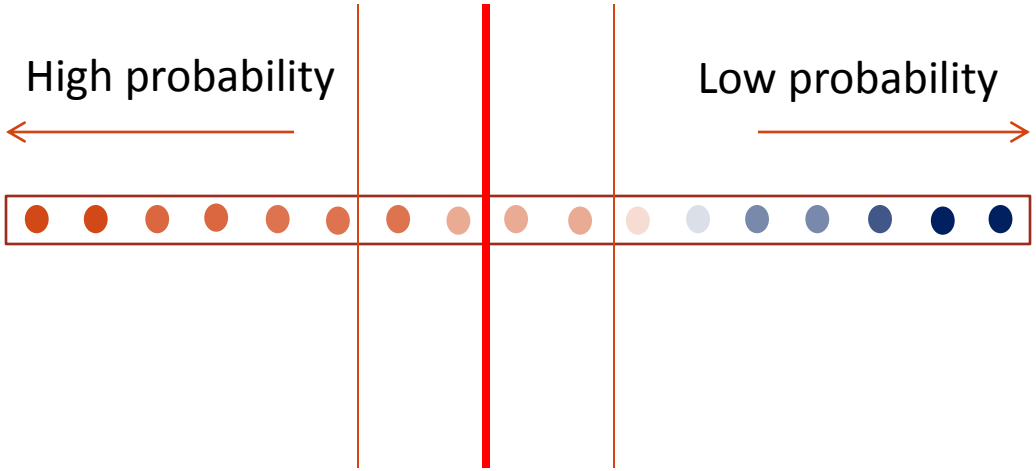


Vector of marginals

Random-Walk-Based Algorithms

Compute marginals of **random walk started at seed**





Output sweep cut of minimum conductance

Random-Walk-Based Algorithms

Algorithm	Output Conductance	Support Size	Running Time
[Spielman, Teng'04]	$\sqrt{\phi(A)\log^3 \mathrm{vol}(A)}$	$O(\mathrm{Vol}(A))$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\phi(A)^2}\right)$
[Andersen, Chung, Lang '07]	$\sqrt{\phi(A)\log \mathrm{vol}(A)}$	$O(\mathrm{Vol}(A))$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\phi(A)}\right)$
[Andersen, Peres '09]	$\sqrt{\phi(A)\log \mathrm{vol}(A)}$	$O(\mathrm{Vol}(A))$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\sqrt{\phi(A)}}\right)$
[Oveis Gharan, Trevisan '12]]	$\sqrt{\frac{\phi(A)}{\epsilon}}$	$O(\mathrm{Vol}(A)^{1+\epsilon})$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\sqrt{\phi(A)}}\right)$

Random-Walk-Based Algorithms

Algorithm	Output Conductance	Support Size	Running Time
[Spielman, Teng'04]	$\sqrt{\phi(A)\log^3 \mathrm{vol}(A)}$	$O(\mathrm{Vol}(A))$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\phi(A)^2}\right)$
[Andersen, Chung, Lang '07]	$\sqrt{\phi(A)\log \mathrm{vol}(A)}$	$O(\mathrm{Vol}(A))$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\phi(A)}\right)$
[Andersen, Peres '09]	$\sqrt{\phi(A)\log \mathrm{vol}(A)}$	$O(\mathrm{Vol}(A))$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\sqrt{\phi(A)}}\right)$
[Oveis Gharan, Trevisan '12]]	$\sqrt{\frac{\phi(A)}{\epsilon}}$	$O(\mathrm{Vol}(A)^{1+\epsilon})$	$\tilde{O}\left(\frac{\mathrm{Vol}(A)}{\sqrt{\phi(A)}}\right)$

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 τ_A of graph induced by target cut A.

Well-Connected Local Clustering

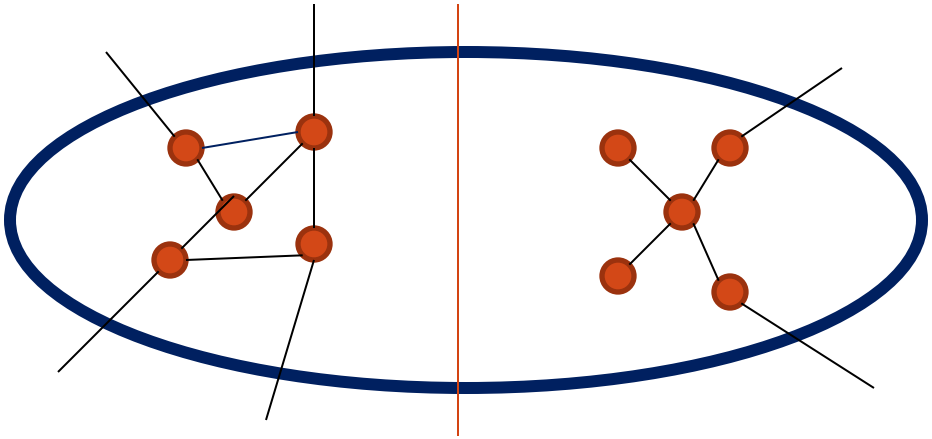
OBSERVATION:
Spectral, random-walk-based algorithms **in practice work better than theoretical bounds**

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 τ_A of graph induced by target cut A.



Input cluster A
should be well connected

Well-Connected Local Clustering

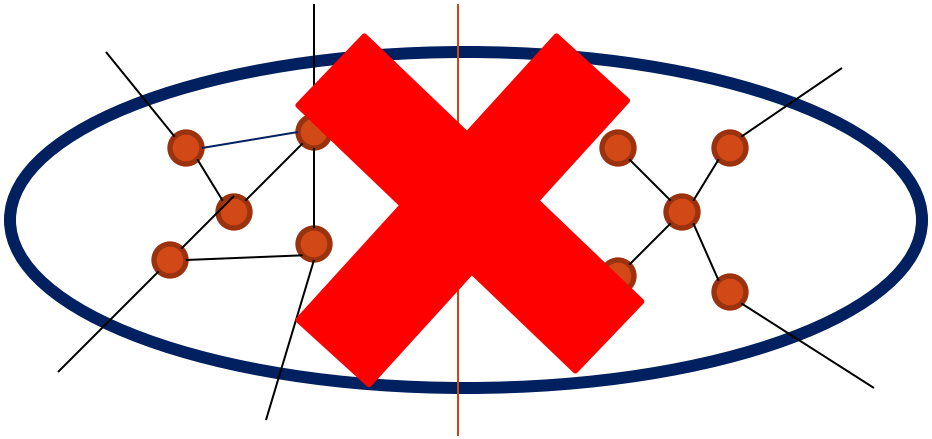
OBSERVATION:
Spectral, random-walk-based algorithms **in practice work better than theoretical bounds**

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 τ_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 τ_A of graph induced by target cut A.

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

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

MIXING TIME WITHIN SET A
SMALLER THAN TO THE OUTSIDE

The random-walk-based algorithm outputs cut S with

$$\phi(S) \cdot \underbrace{O(\sqrt{\tau_A})}_{\text{APPROXIMATION RATIO}} \cdot \phi(A) \quad \text{and} \quad \mathrm{vol}(\mathrm{Explored})) \cdot O(\mathrm{vol}(A))$$

LOCALITY GUARANTEE

Limitations of Random-Walks

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

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

MIXING TIME WITHIN SET A SMALLER THAN TO THE OUTSIDE

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

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

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

Limitations of Random-Walks

[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

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

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

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

Typical Cheeger-like spectral guarantee

Our Result

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

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

MIXING TIME WITHIN SET A
SMALLER THAN TO THE OUTSIDE

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

$$\phi(S) \cdot \quad O(\phi(A))$$ and $$\mathrm{vol}(\mathrm{Explored})) \cdot \quad O(\mathrm{vol}(A))$$

Our Result

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

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

MIXING TIME WITHIN SET A
SMALLER THAN TO THE OUTSIDE

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

$$\underbrace{\phi(S) \cdot O(\phi(A))}_{\text{CONSTANT APPROXIMATION}} \quad \text{and} \quad \underbrace{\mathrm{vol}(\mathrm{Explored})) \cdot O(\mathrm{vol}(A))}_{\text{CONSTANT SIZE BLOW-UP}}$$

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

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

OVERLAP PROPERTY

Overlap Property

ADDITIONAL PROPERTY OF RANDOM-WALK –BASED ALGORITHM:

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

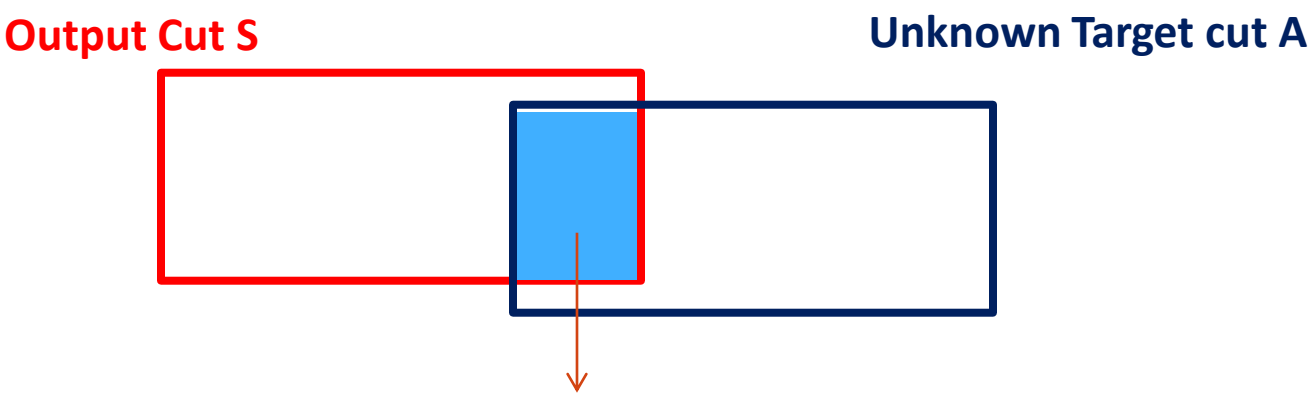
$$\tau_A \cdot O\left(\frac{1}{\phi(A)}\right)$$

MIXING TIME WITHIN SET A SMALLER THAN TO THE OUTSIDE

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

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

OVERLAP PROPERTY



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

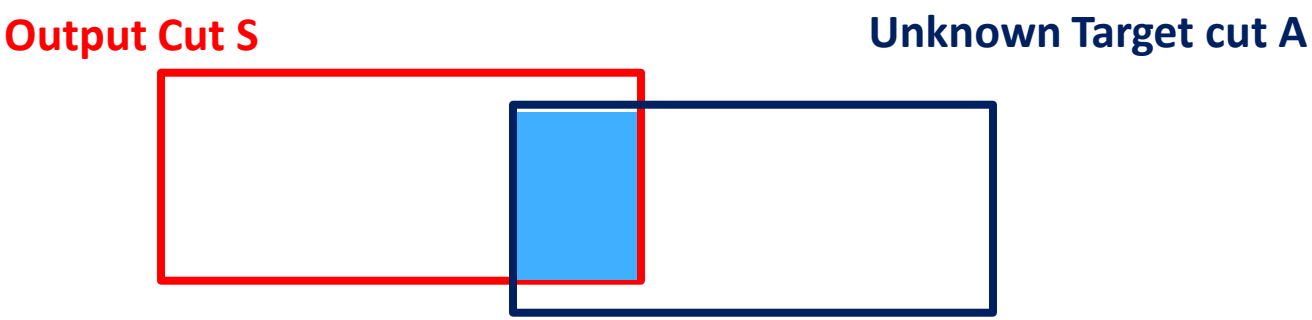
$$\tau_A \cdot \; O\left(\frac{1}{\phi(A)}\right)$$

MIXING TIME WITHIN SET A SMALLER THAN TO THE OUTSIDE

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

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

OVERLAP PROPERTY



GOAL: Exploit overlap to produce better approximation to ϕ(A)

Cut-Improvement Algorithms

[Lang, Rao '93][Andersen, Lang '08]

OVERLAP GUARANTEE → APPROXIMATION GUARANTEE

Cut-Improvement Algorithms

[Lang, Rao '93][Andersen, Lang '08]

OVERLAP GUARANTEE $\longrightarrow$ APPROXIMATION GUARANTEE

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

$$\frac{\mathrm{vol}(S \cap A)}{\mathrm{vol}(A)} \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} + \delta \frac{\mathrm{vol}(\bar{S})}{\mathrm{vol}(V)}$$

OVERLAP GUARANTEE

we have

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

APPROXIMATION GUARANTEE

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

$$\frac{\mathrm{vol}(S \cap A)}{\mathrm{vol}(A)} \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} + \delta \frac{\mathrm{vol}(\bar{S})}{\mathrm{vol}(V)}$$

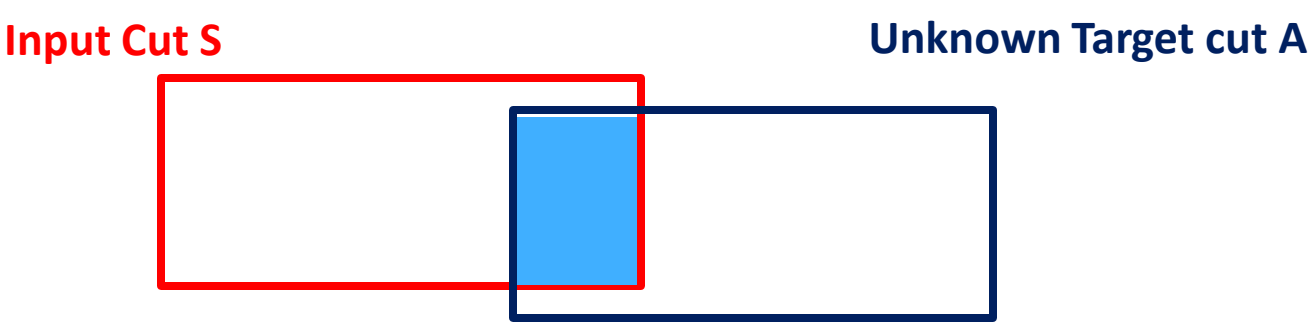
OVERLAP GUARANTEE

we have

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

APPROXIMATION GUARANTEE

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



Cut-Improvement Algorithms

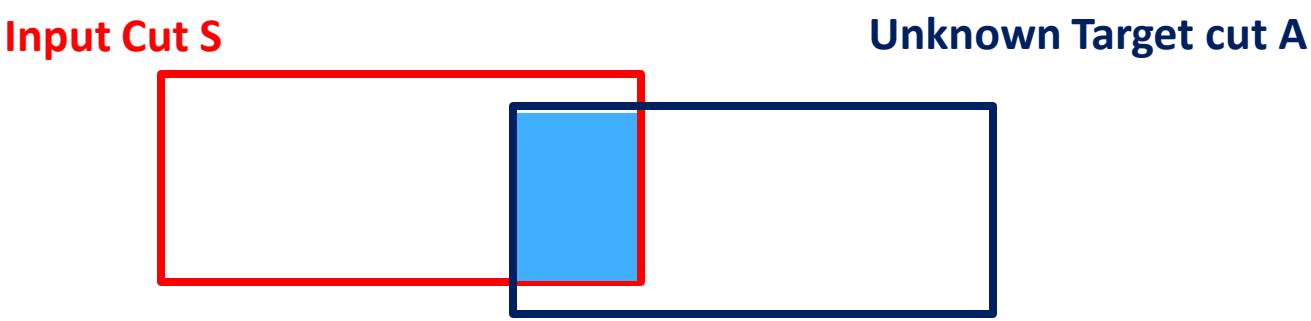
[Lang, Rao '93][Andersen, Lang '08]
On input cut S, this algorithm outputs cut C such that, for any target cut A with

$$\frac{\mathrm{vol}(S \cap A)}{\mathrm{vol}(A)} \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} + \delta \frac{\mathrm{vol}(\bar{S})}{\mathrm{vol}(V)} \boxed{\geq \delta \geq \square(1)}$$

we have

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

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



IN OUR CASE: $\mathrm{vol}(V), \mathrm{vol}(\bar{S}) \to \infty$

Cut-Improvement Algorithms

[Lang, Rao '93][Andersen, Lang '08]

OVERLAP GUARANTEE $\longrightarrow$ APPROXIMATION GUARANTEE

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

$$\frac{\mathrm{vol}(S \cap A)}{\mathrm{vol}(A)} \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} + \delta \frac{\mathrm{vol}(\bar{S})}{\mathrm{vol}(V)} \boxed{\geq \delta \geq \square(1)}$$

we have

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

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

IN OUR CASE: $\mathrm{vol}(V), \mathrm{vol}(\bar{S}) \to \infty$

Cut-Improvement Algorithms

[Lang, Rao '93][Andersen, Lang '08]

OVERLAP GUARANTEE $\longrightarrow$ APPROXIMATION GUARANTEE

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

$$\frac{\mathrm{vol}(S \cap A)}{\mathrm{vol}(A)} \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} + \delta \frac{\mathrm{vol}(\bar{S})}{\mathrm{vol}(V)} \boxed{\geq \delta \geq \square(1)}$$

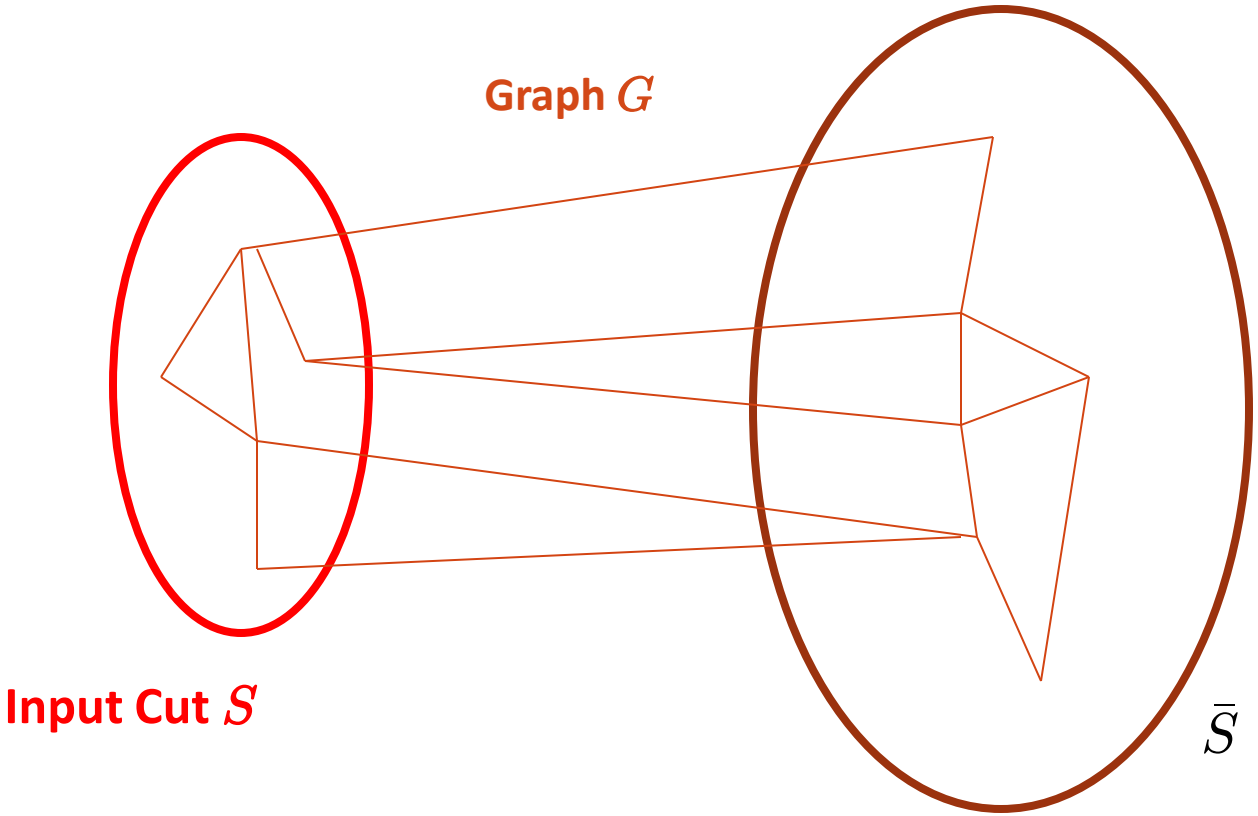
we have

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

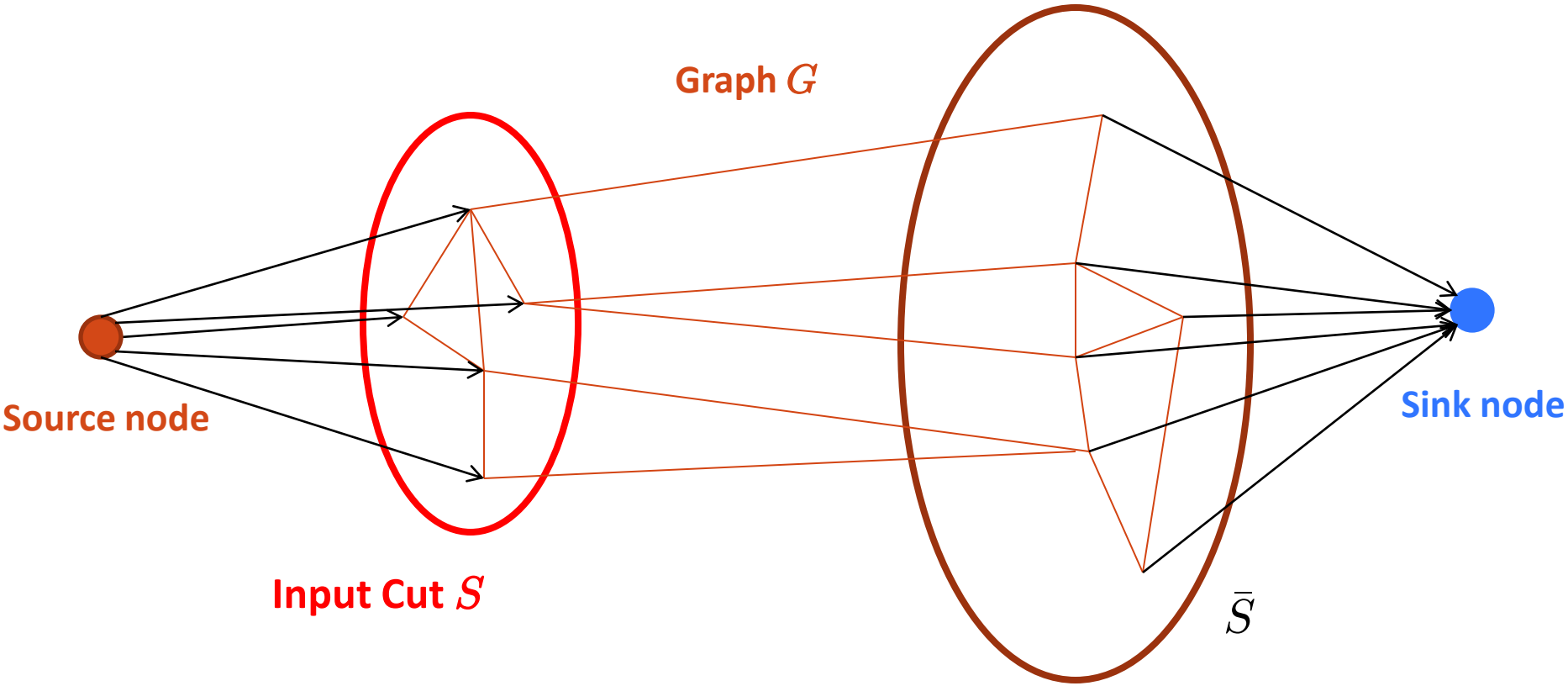
The algorithm runs a small number of **global s-t maxflow computations**.

IN OUR CASE: $\mathrm{vol}(V), \mathrm{vol}(\bar{S}) \to \infty$

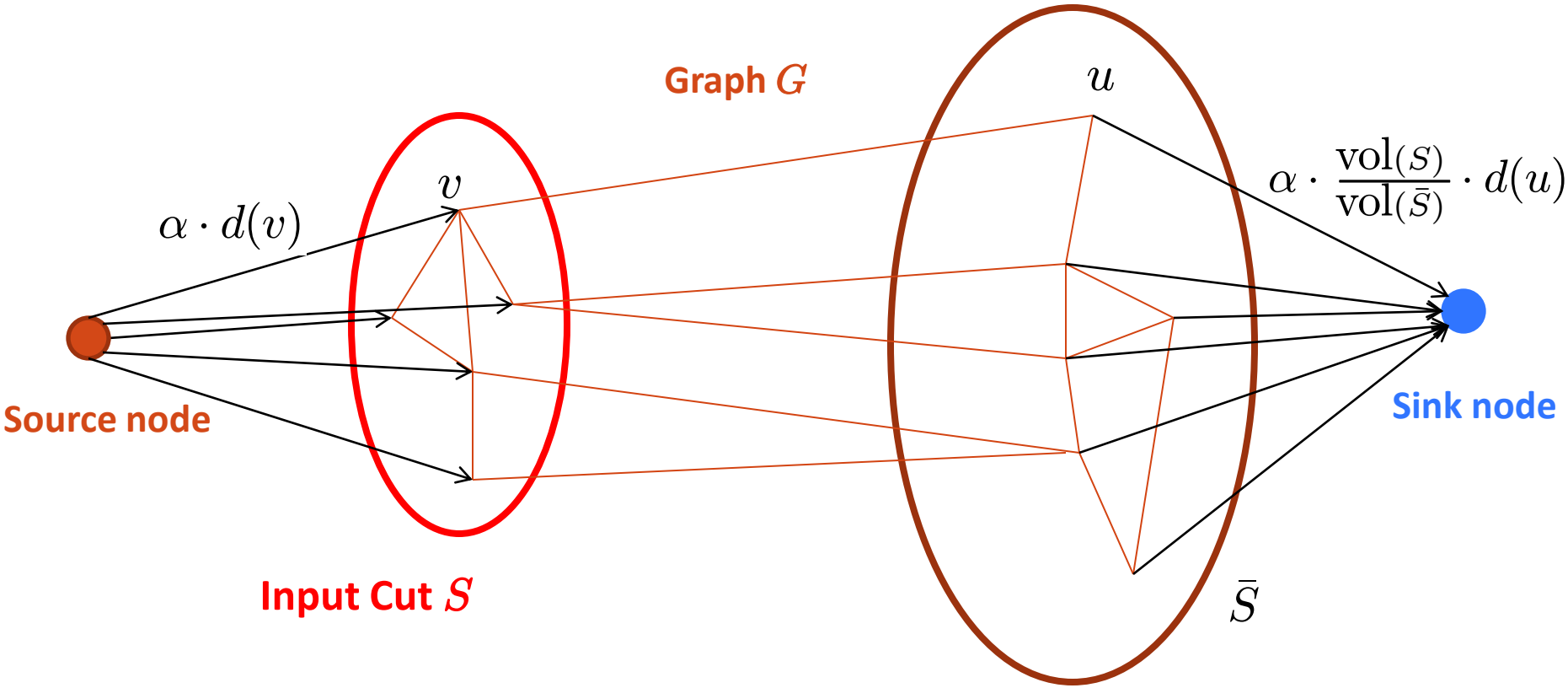
Global Solution to Cut-Improvement



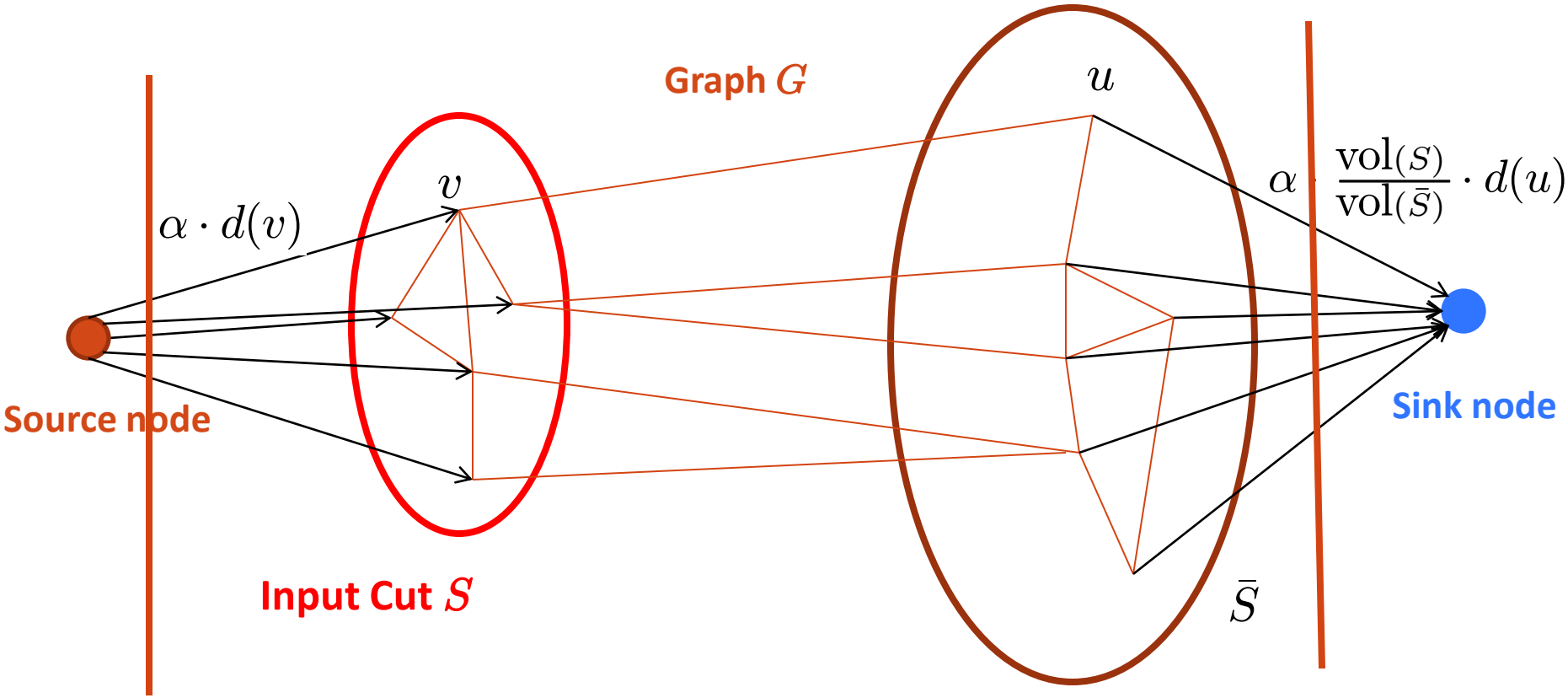
Global Solution to Cut-Improvement



Global Solution to Cut-Improvement



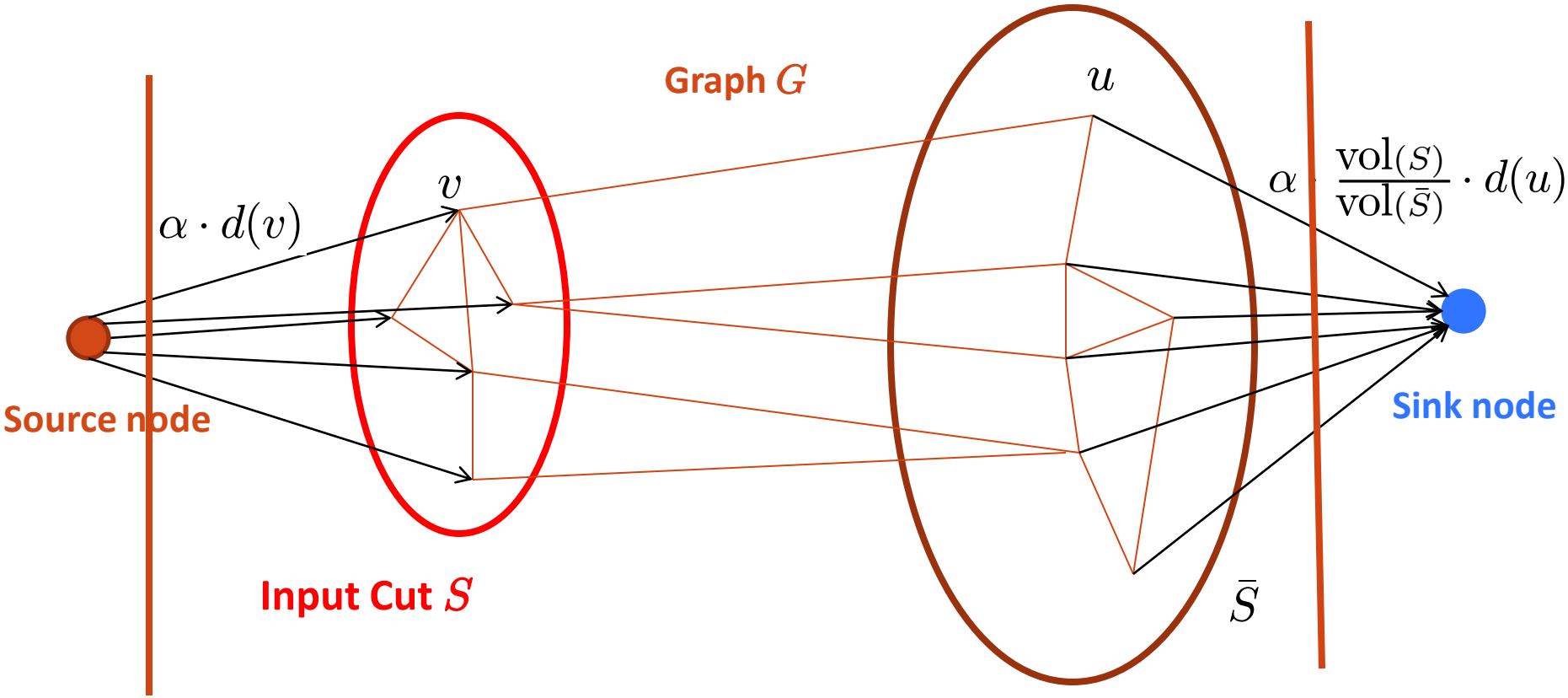
Global Solution to Cut-Improvement



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

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

Global Solution to Cut-Improvement



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

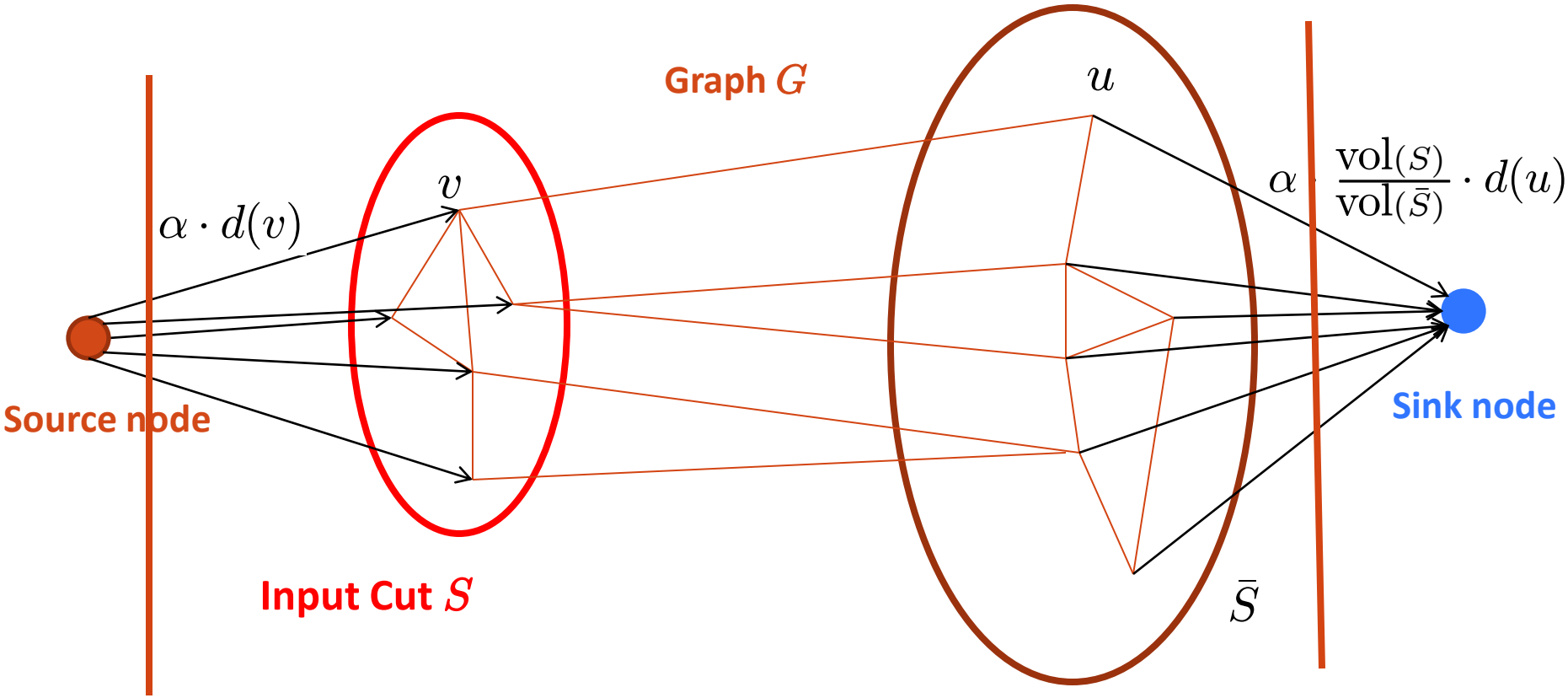
Cut capacity = $\alpha \ \mathrm{vol}(S)$

Solve parametric flow problem:

$$\max \ \alpha$$

s.t. total source/sink capacity can be routed

Global Solution to Cut-Improvement

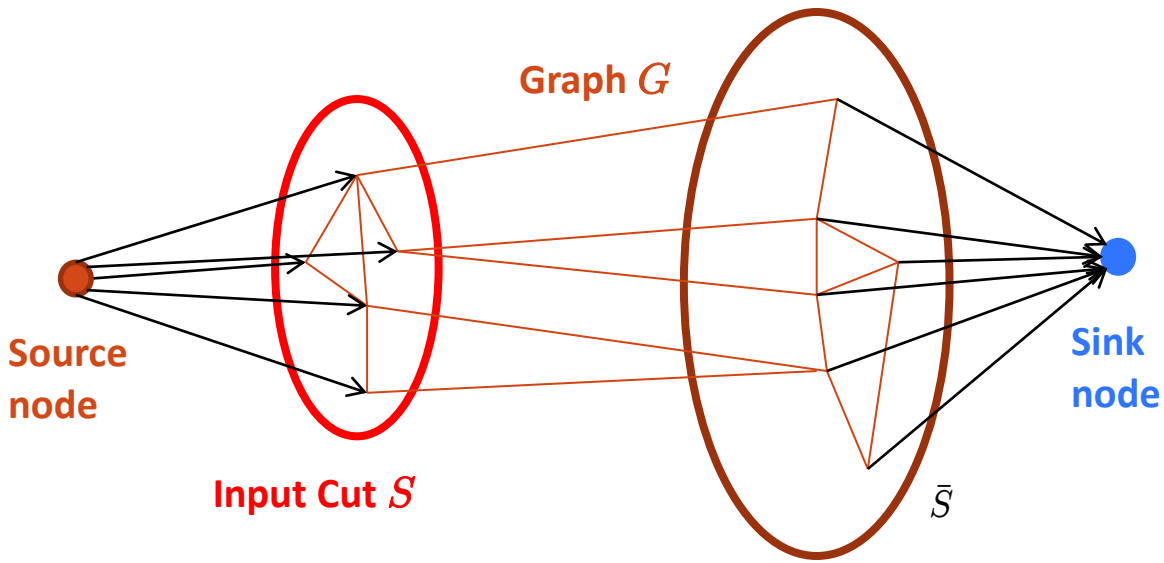


Cut capacity = $\alpha \ \mathrm{vol}(S)$

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

PROBLEM: optimal (and nearly-optimal) flow solutions are **inherently global**

Localization of Cut-Improvement

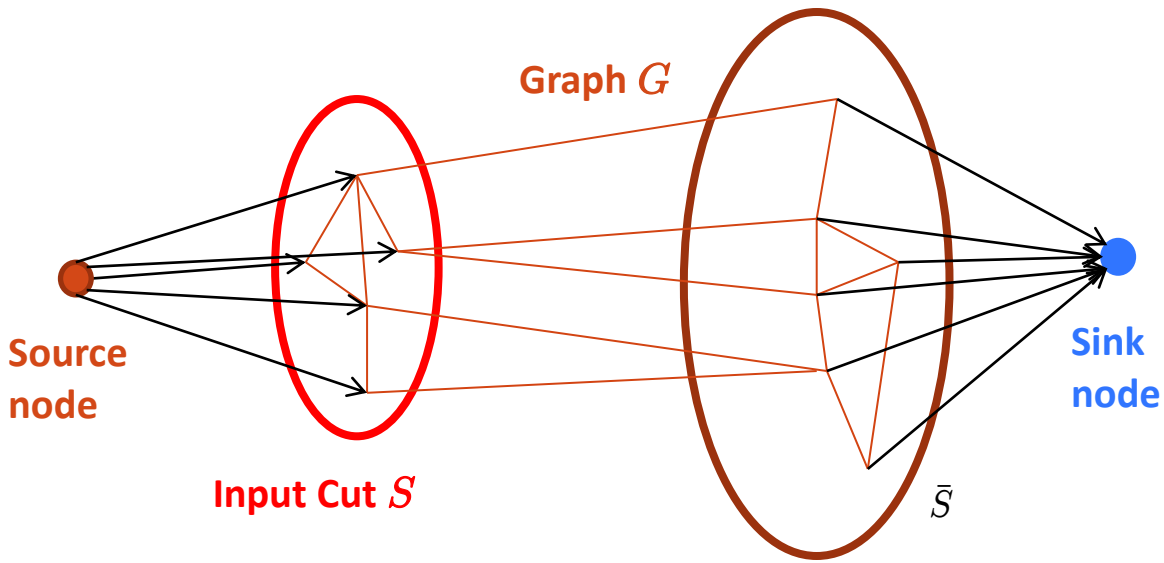


QUESTION: Can the cut-improvement algorithm be localized?

PESSIMISTIC CONJECTURE: **NO.**

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

Localization of Cut-Improvement



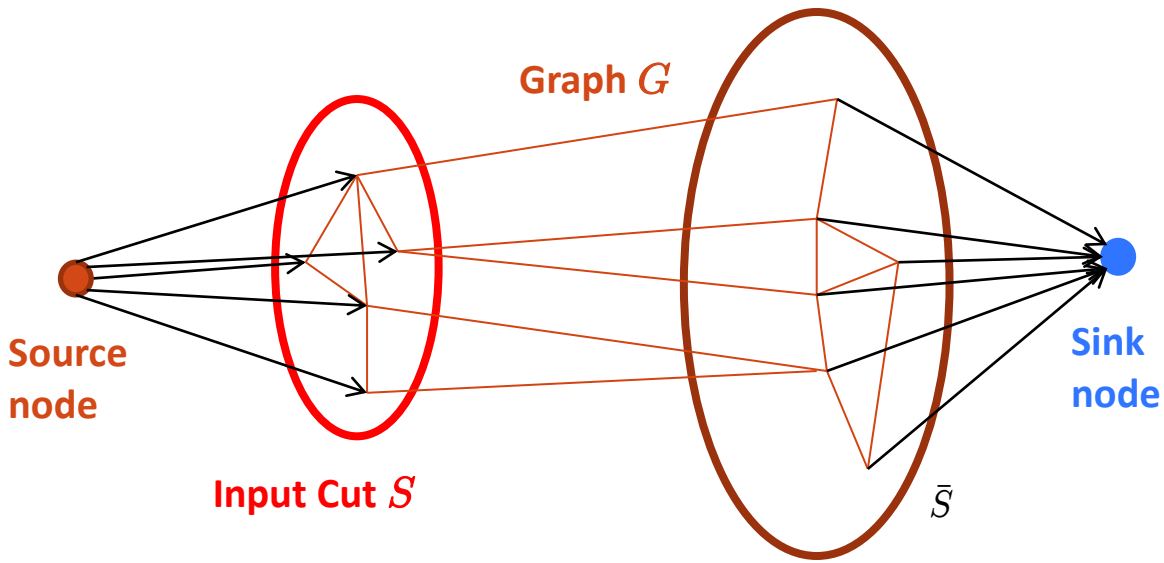
QUESTION: Can the cut-improvement algorithm be localized?

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?

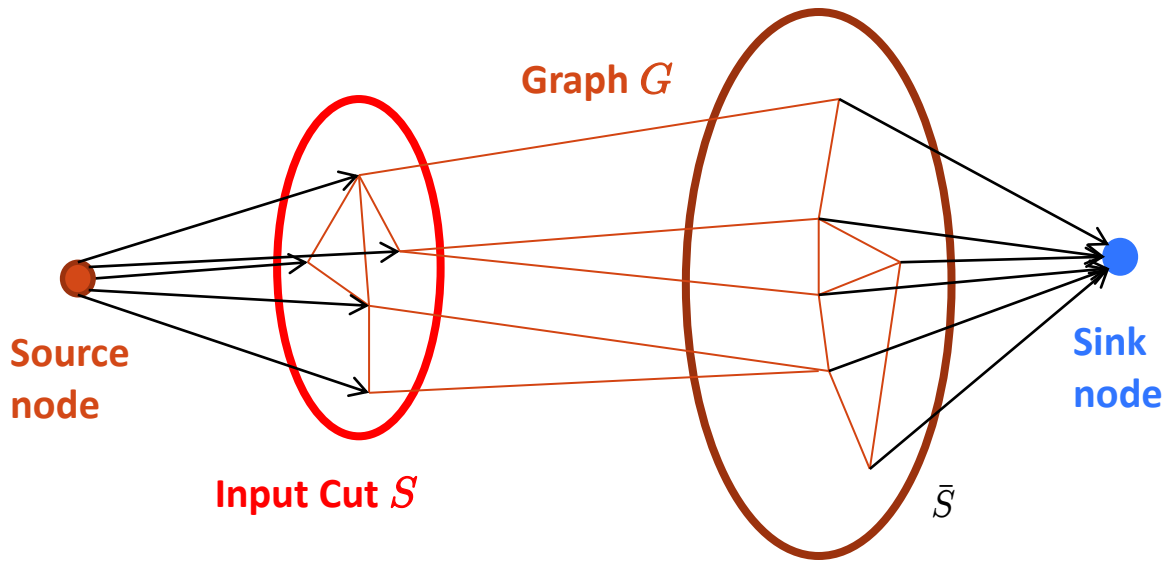
PESSIMISTIC CONJECTURE: **NO.**

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



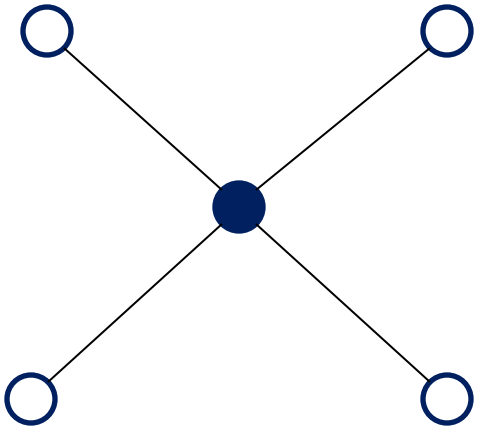
QUESTION: Can the cut-improvement algorithm be localized?

PESSIMISTIC CONJECTURE: **NO.**

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

Localization of Random-Walk

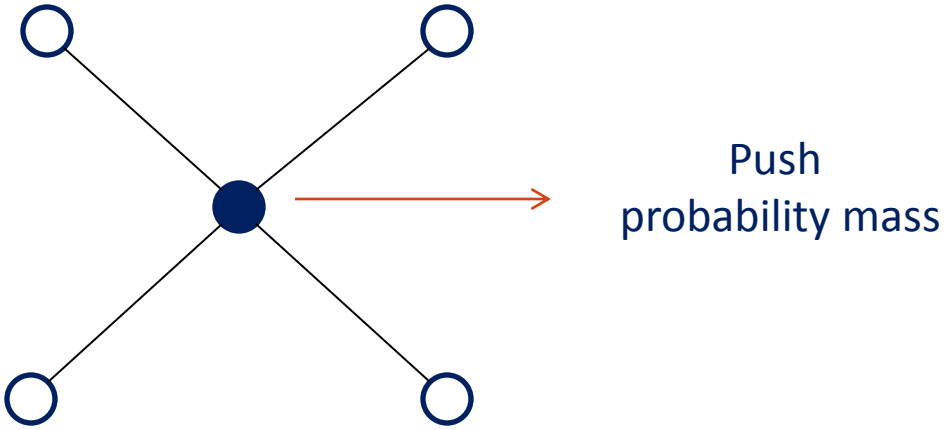
Marginals are computed by ``**pushing'' probability mass** from vertices:



Argument applies to PageRank random walk, but can be generalized to other walks.

Localization of Random-Walk

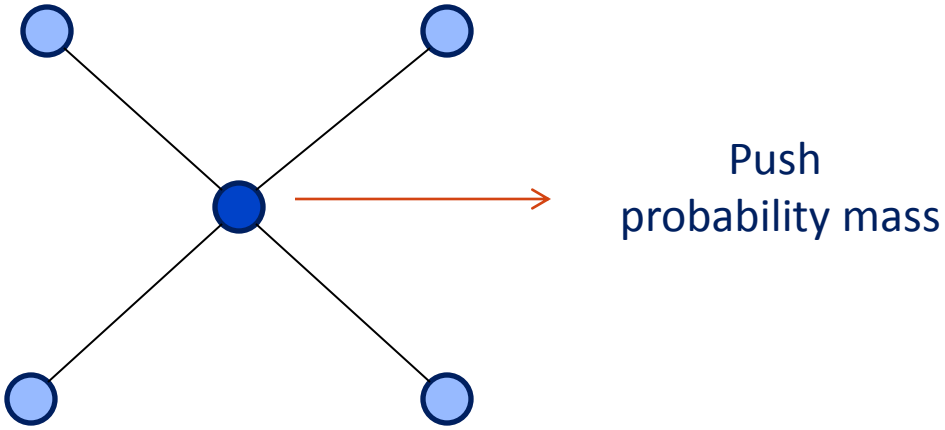
Marginals are computed by ``**pushing'' probability mass** from vertices:



Argument applies to PageRank random walk, but can be generalized to other walks.

Localization of Random-Walk

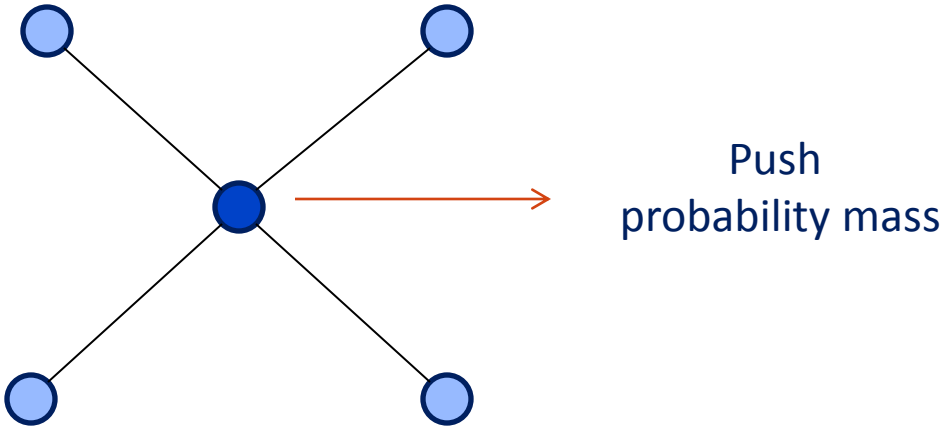
Marginals are computed by ``**pushing'' probability mass** from vertices:



Argument applies to PageRank random walk, but can be generalized to other walks.

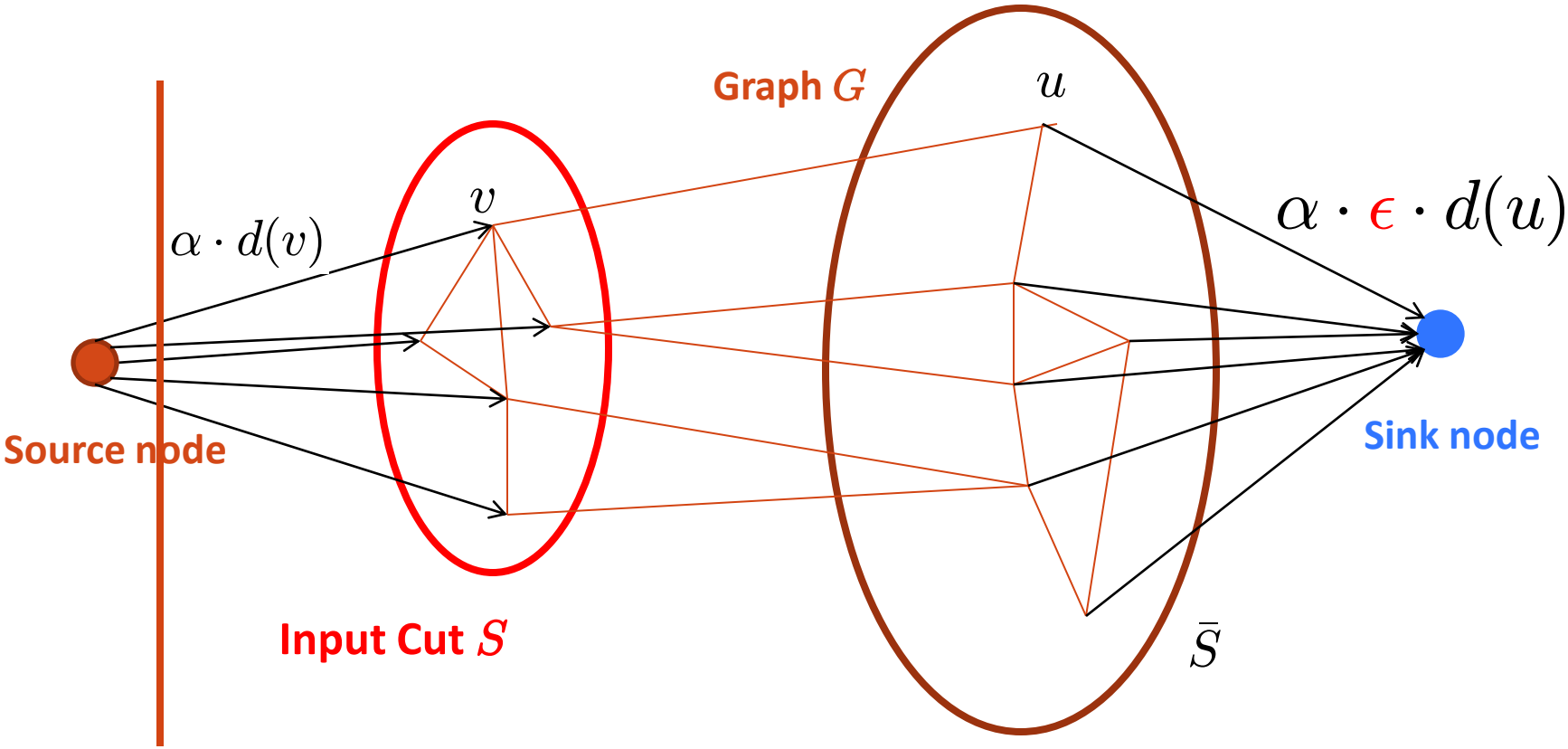
Localization of Random-Walk

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

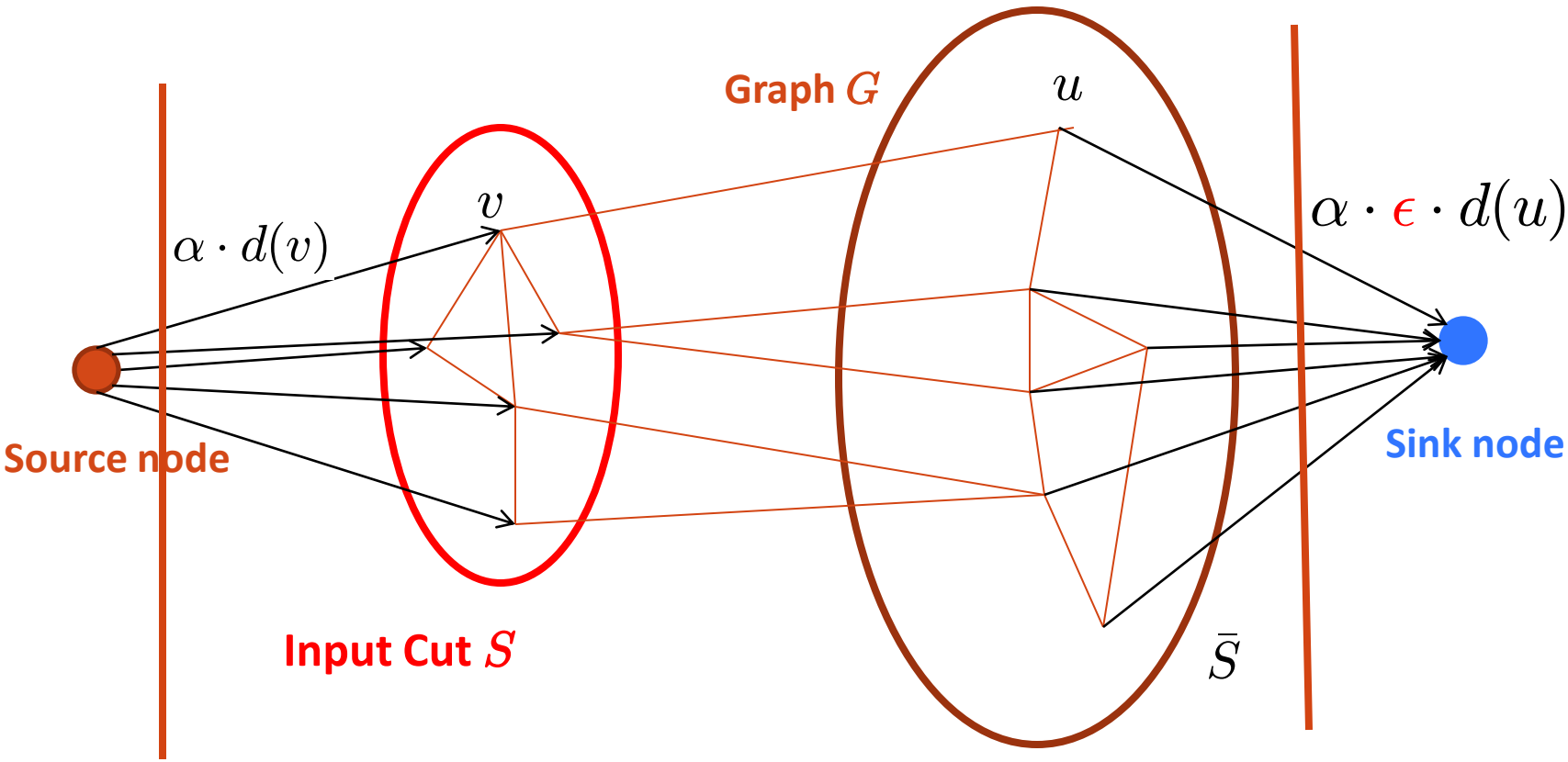
Modification of Flow Problem Achieves Localization



Cut capacity = $\alpha \, \mathrm{vol}(S)$

Every sink-side node has a large capacity to the sink: $\epsilon = \square(1)$

Modification of Flow Problem Achieves Localization

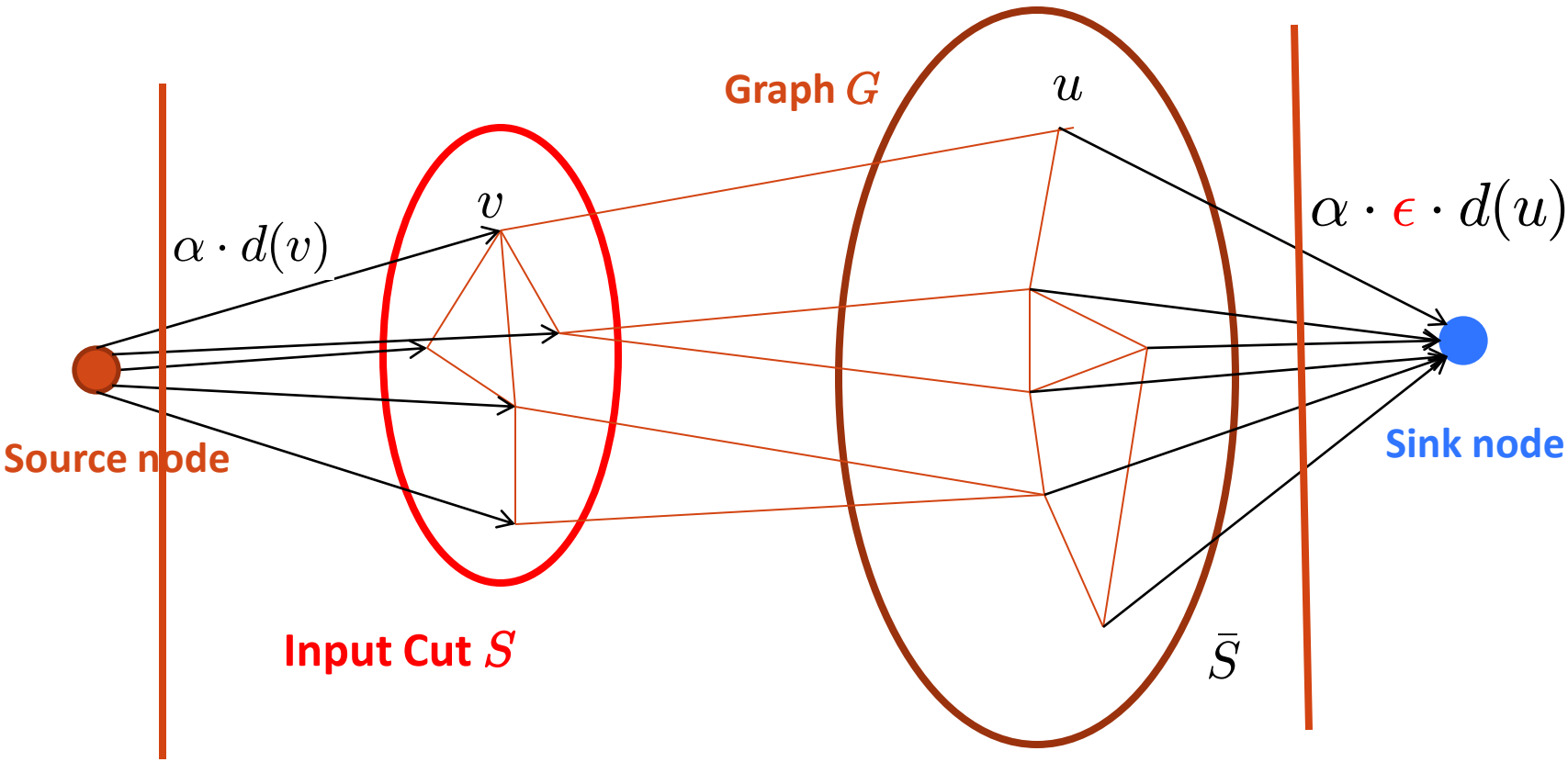


Cut capacity = $\alpha \, \mathrm{vol}(S)$ $\qquad \epsilon = \square(1)$ $\qquad$ Cut capacity = $\alpha \cdot \epsilon \cdot \mathrm{vol}(\bar{S})$

CONSEQUENCE: There exist local optimal solutions:

$$\mathrm{vol}(\mathrm{Explored}) \cdot \frac{\alpha \mathrm{vol}(S)}{\alpha \epsilon} = \frac{\mathrm{vol}(S)}{\epsilon}$$

Modification of Flow Problem Achieves Localization



Cut capacity = $\alpha \, \mathrm{vol}(S)$ $\qquad \epsilon = \square(1) \qquad$ Cut capacity = $\alpha \cdot \epsilon \cdot \mathrm{vol}(\bar{S})$

CONSEQUENCE: There exist local optimal solutions: $\mathrm{vol}(\mathrm{Explored}) \cdot \dfrac{\mathrm{vol}(S)}{\epsilon}$

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

Optimization Interpretation

GLOBAL CUT-IMPROVEMENT on input S:

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

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

Optimization Interpretation

GLOBAL CUT-IMPROVEMENT on input S:

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

CORRELATION BETWEEN C AND S

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

Optimization Interpretation

GLOBAL CUT-IMPROVEMENT on input S:

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

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

LOCALIZED CUT-IMPROVEMENT – RESTRICT FEASIBLE SETS C:

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

$$\boxed{\text{s.t. } \frac{\mathrm{vol}(C \cap S)}{\mathrm{vol}(C)} \geq \square(\epsilon).}$$

OVERLAP LOWER BOUND

Optimization Interpretation

LOCALIZED CUT-IMPROVEMENT – RESTRICT FEASIBLE SETS C:

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

$$\boxed{\text{s.t. } \frac{\mathrm{vol}(C \cap S)}{\mathrm{vol}(C)} \geq \square(\epsilon).}$$

OVERLAP LOWER BOUND

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!