Approximate Gomory-Hu Tree in ~Max-Flow Time

Jason Li (CMU)

joint work with Debmalya Panigrahi (Duke)

STOC 2021

Introduction

This talk: all graphs are unweighted and undirected

Introduction

This talk: all graphs are unweighted and undirected

Gomory-Hu tree: a tree on same vertex set s.t. $\text{mincut}_T(u,v)=\text{mincut}_G(u,v)$

- Can be computed using n-1 max-flows [Gomory-Hu'61]
- No faster algorithm for general graphs!
- Unweighted graphs: $\tilde{O}(mn)$ time [BHKP'07], $m^{3/2}n^{1/6}$ [AKT SODA'20]

Introduction

This talk: all graphs are unweighted and undirected

Gomory-Hu tree: a tree on same vertex set s.t. $\text{mincut}_T(u,v)=\text{mincut}_G(u,v)$

- Can be computed using n-1 max-flows [Gomory-Hu'61]
- No faster algorithm for general graphs!
- Unweighted graphs: $\tilde{O}(mn)$ time [BHKP'07], $m^{3/2}n^{1/6}$ [AKT SODA'20]

Single-source mincut (SSMC): given source vertex s, find all mincut(s,v)

- Can be used to compute GH tree [AKT FOCS'20]

Introduction

This talk: all graphs are unweighted and undirected

Gomory-Hu tree: a tree on same vertex set s.t. $\text{mincut}_T(u,v)=\text{mincut}_G(u,v)$

- Can be computed using n-1 max-flows [Gomory-Hu'61]
- No faster algorithm for general graphs!
- Unweighted graphs: $\tilde{O}(mn)$ time [BHKP'07], $m^{3/2}n^{1/6}$ [AKT SODA'20]

Single-source mincut (SSMC): given source vertex s, find all mincut(s,v)

- Can be used to compute GH tree [AKT FOCS'20]

This talk: approximations

- Approximate all-pairs mincut in $\tilde{O}(n^2)$ time [AKT FOCS'20]
- Nothing better known for GH tree

This work: $(1+\varepsilon)$-approximate GH tree and SSMC in exact max-flow time

Outline

New primitive: Cut Threshold algorithm

Outline

New primitive: Cut Threshold algorithm

Given s and λ ("cut threshold"),
output all $v \in V$ with mincut(s,v) $\le \lambda$

Outline

New primitive: Cut Threshold algorithm

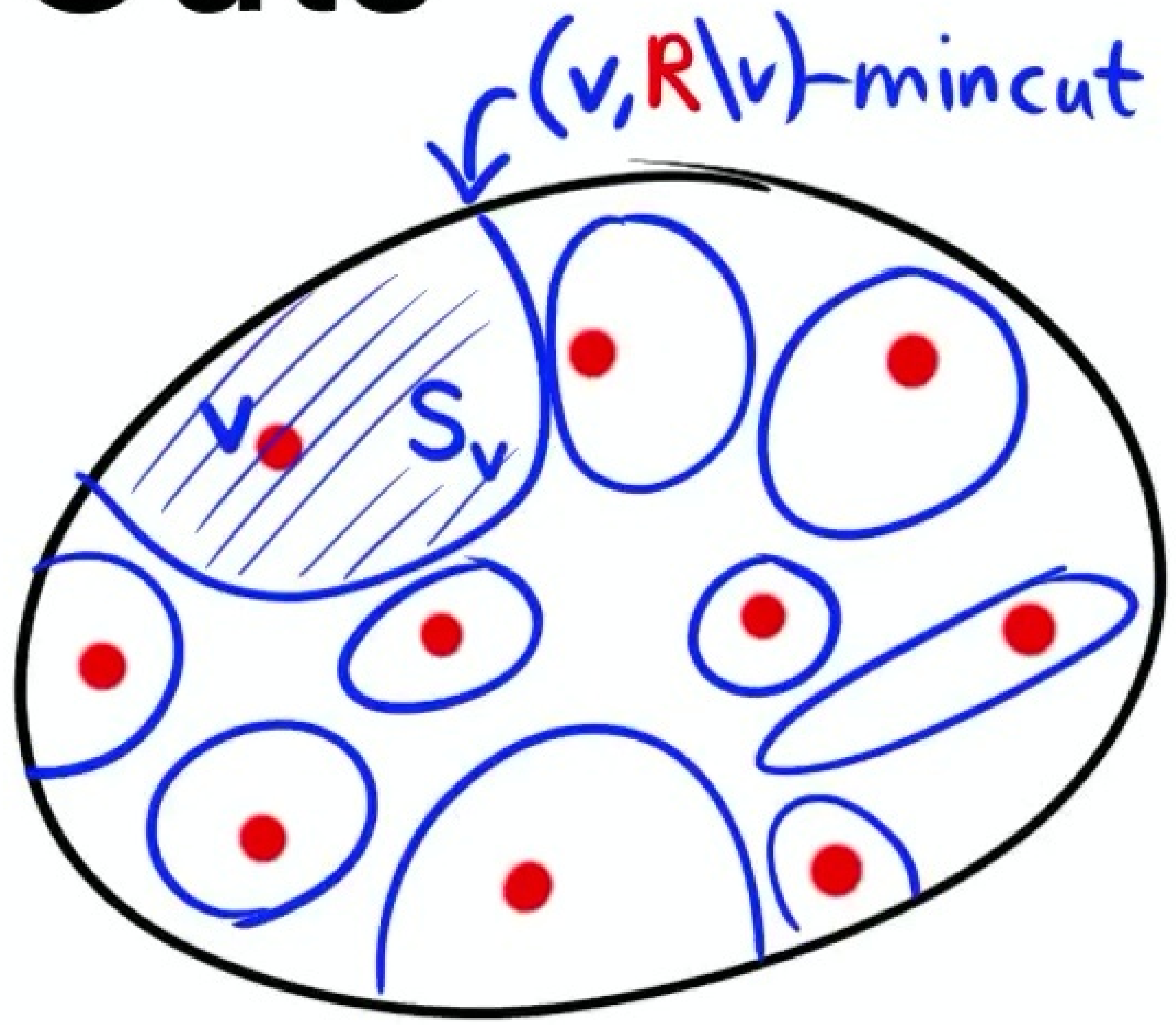
Given s and λ ("cut threshold"),
output all $v \in V$ with $\text{mincut}(s,v) \leq \lambda$

Cut Threshold => approximate single-source mincut
(=> approximate all-pairs mincut [AKT20])

Cut Threshold => approximate Gomory-Hu tree

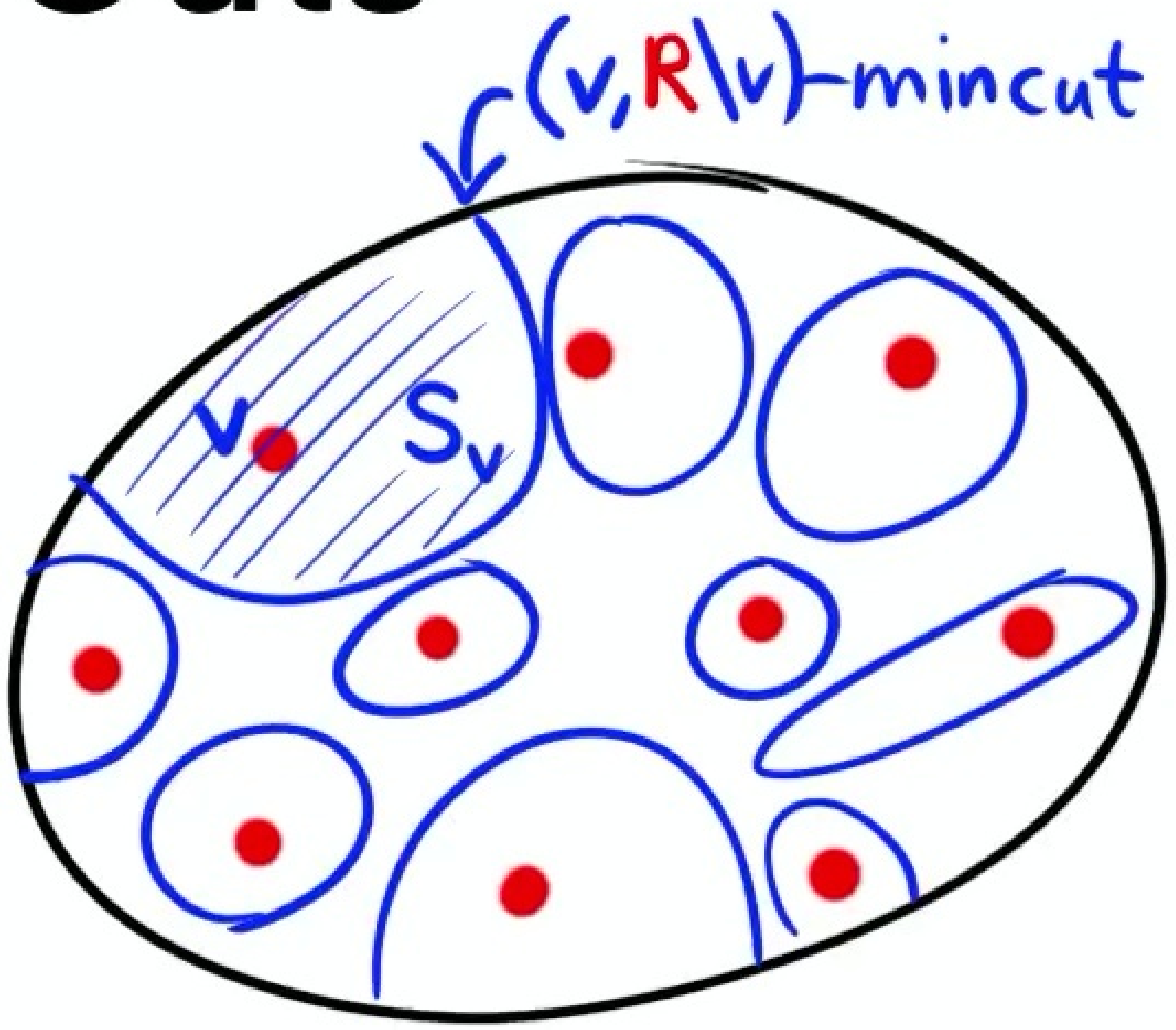
Minimum Isolating Cuts

Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal



Minimum Isolating Cuts

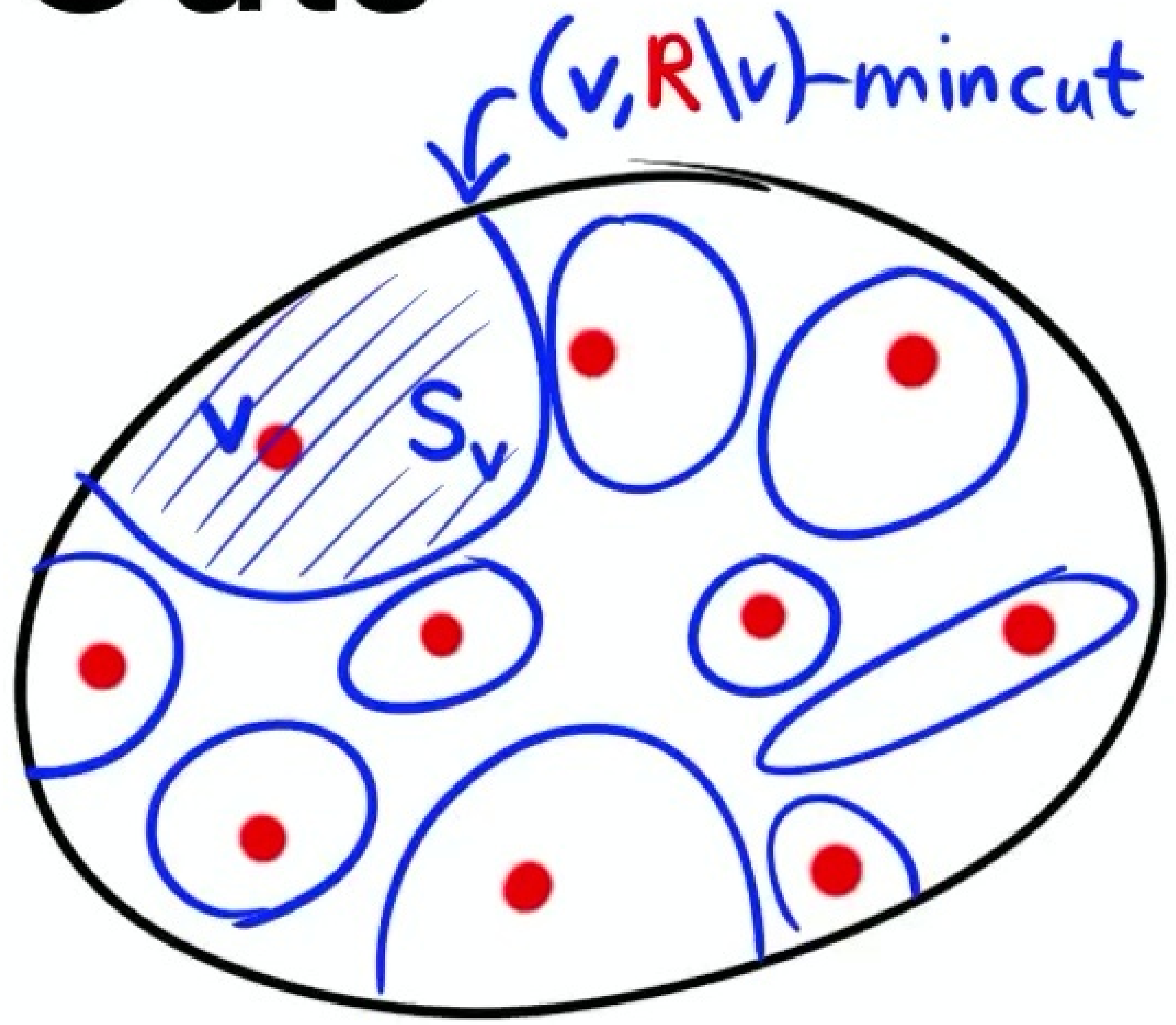
Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal



Trivial: |R| s-t mincuts

[L.-Panigrahi '20] O(log |R|) s-t mincuts suffice!

Minimum Isolating Cuts



Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal

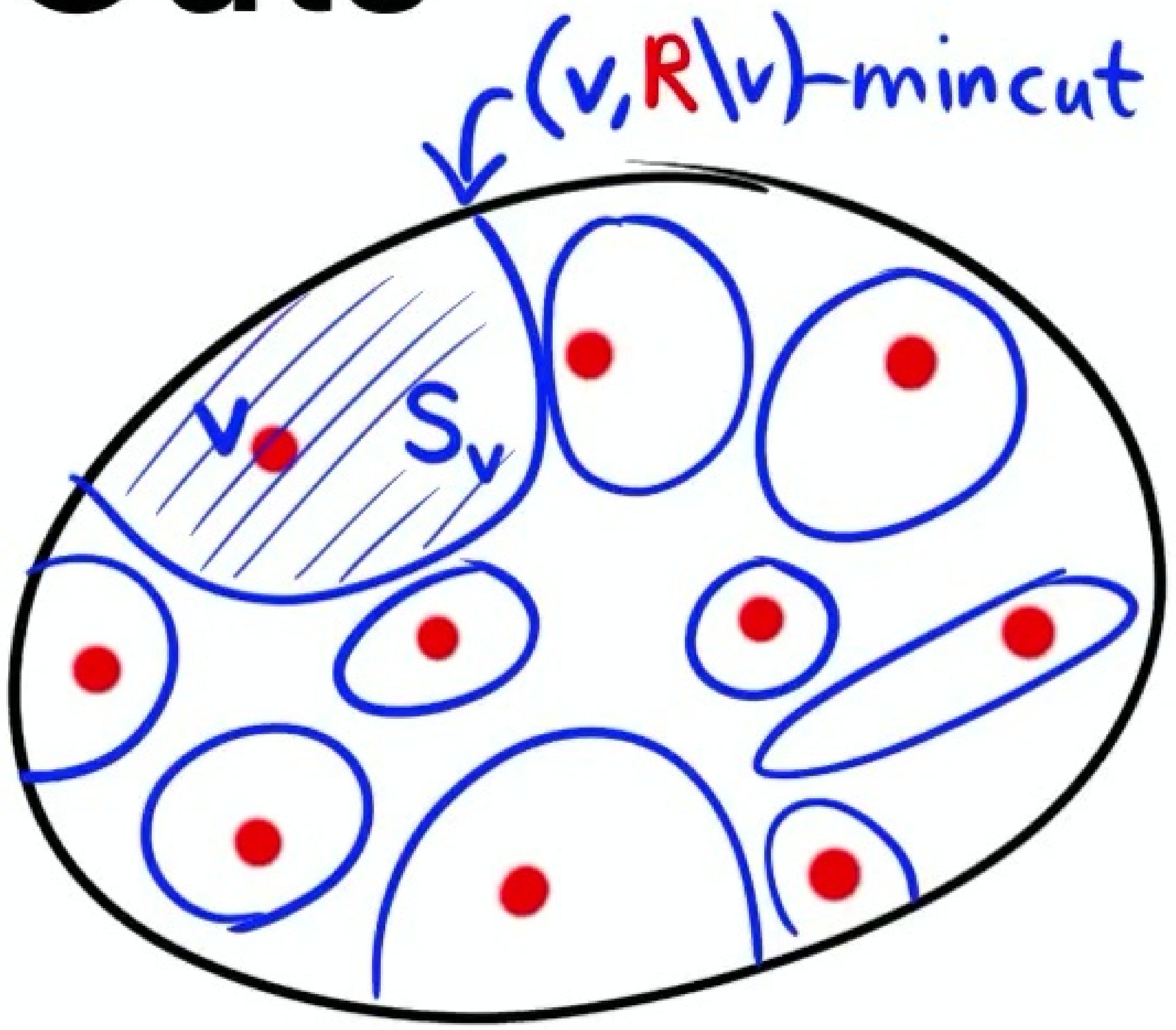
Trivial: |R| s-t mincuts

[L.-Panigrahi '20] O(log |R|) s-t mincuts suffice!

Warm-up: global Steiner mincut (separate at least two terminals)

Minimum Isolating Cuts

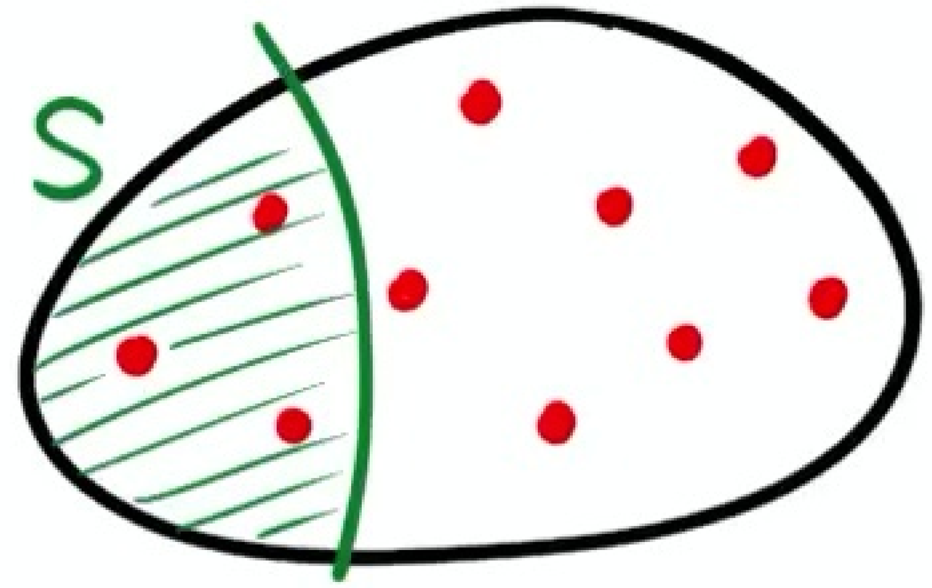
Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal



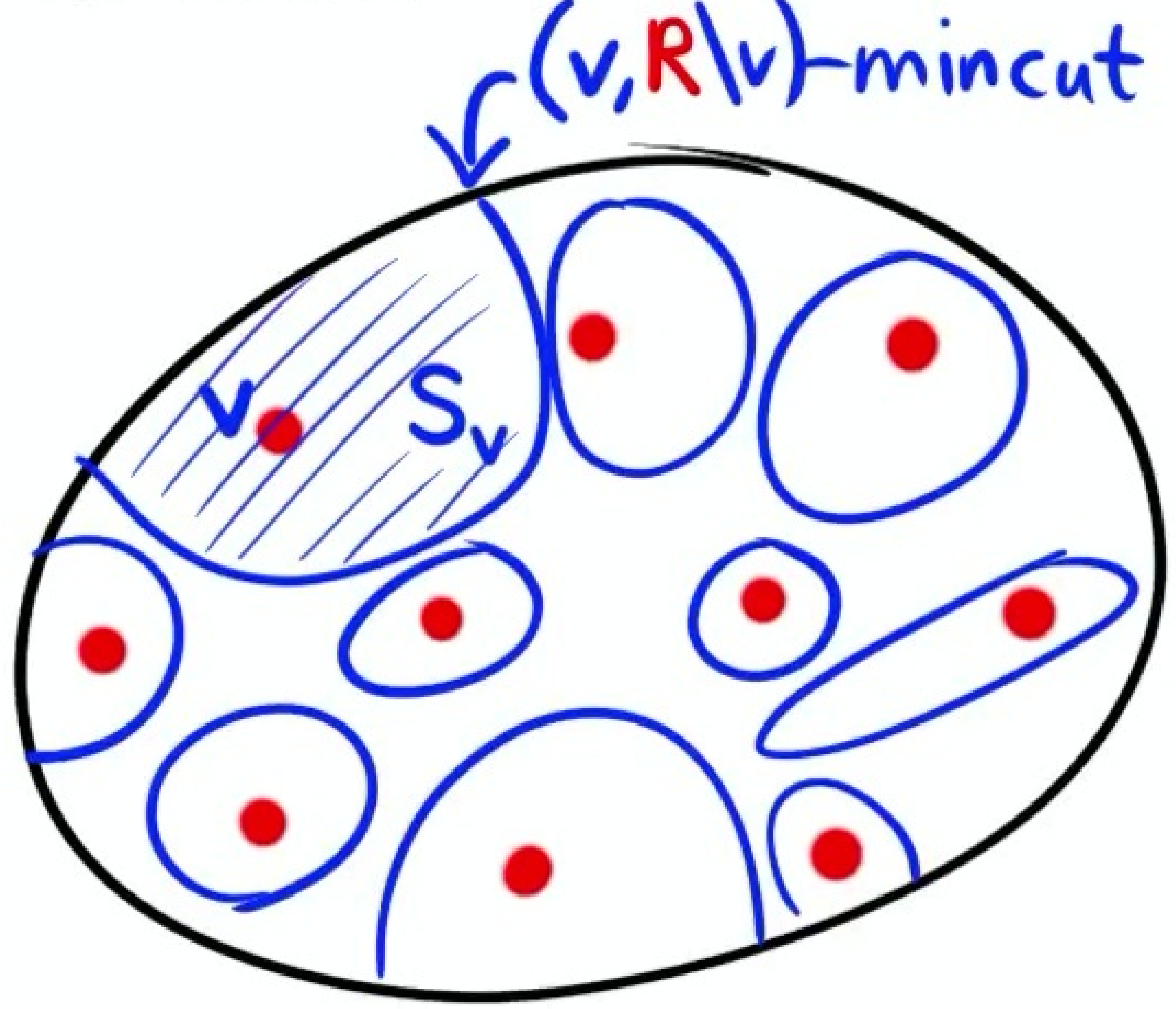
Trivial: |R| s-t mincuts

[L.-Panigrahi '20] O(log |R|) s-t mincuts suffice!

Warm-up: global Steiner mincut (separate at least two terminals)



Minimum Isolating Cuts

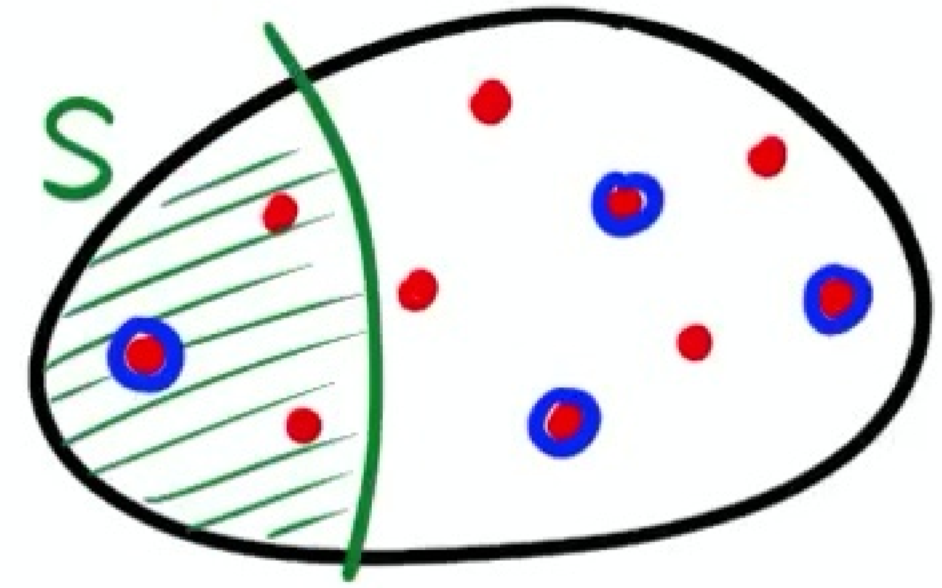


Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal

Trivial: |R| s-t mincuts

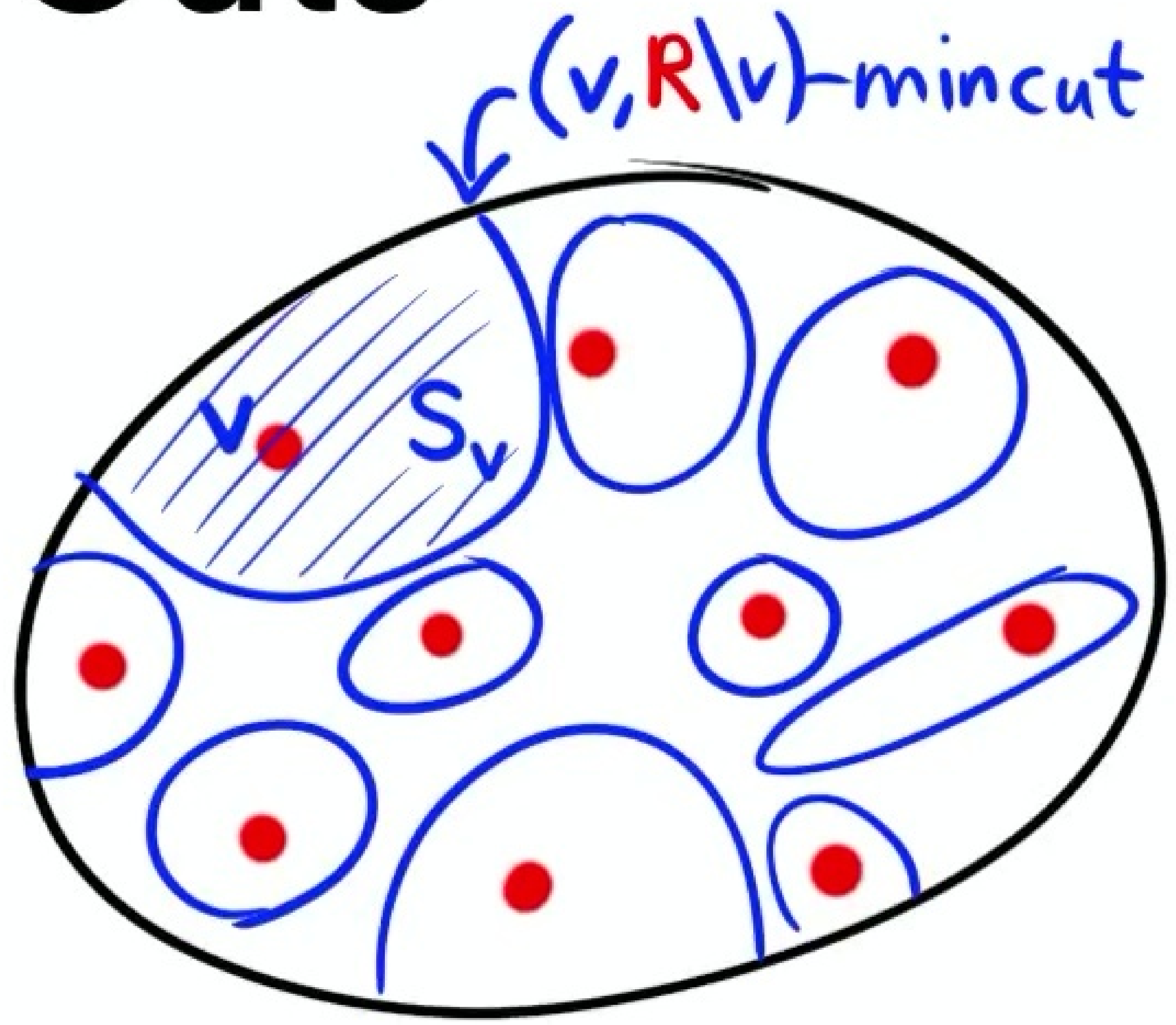
[L.-Panigrahi '20] O(log |R|) s-t mincuts suffice!

Warm-up: global Steiner mincut (separate at least two terminals)



Minimum Isolating Cuts

Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal

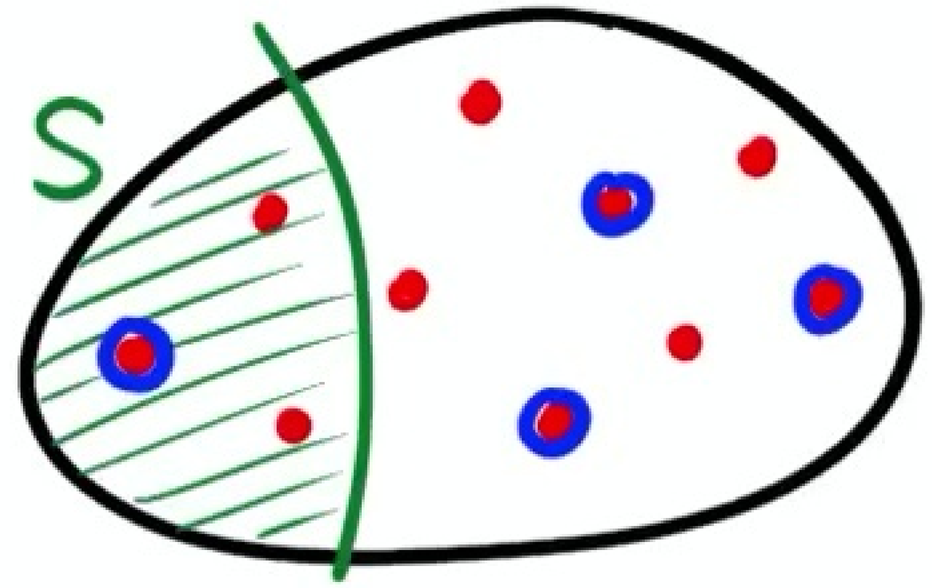


Trivial: |R| s-t mincuts

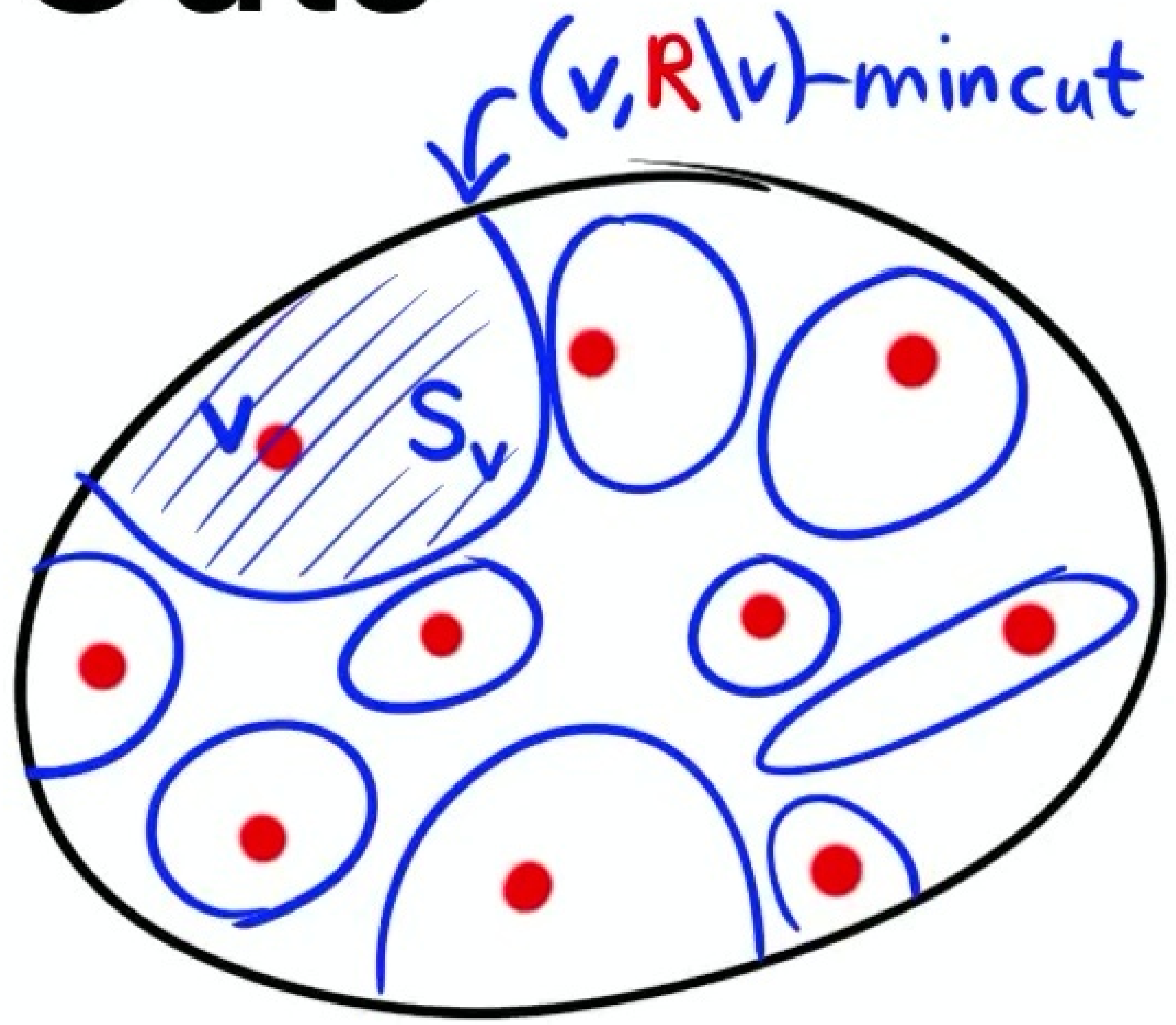
[L.-Panigrahi '20] O(log |R|) s-t mincuts suffice!

Warm-up: global Steiner mincut (separate at least two terminals)

If sample at rate ~ $\frac{1}{|S \cap R|}$, then constant prob. success



Minimum Isolating Cuts



Given a graph and a set R of terminals, find, for each terminal v, the mincut S_v that isolates that terminal

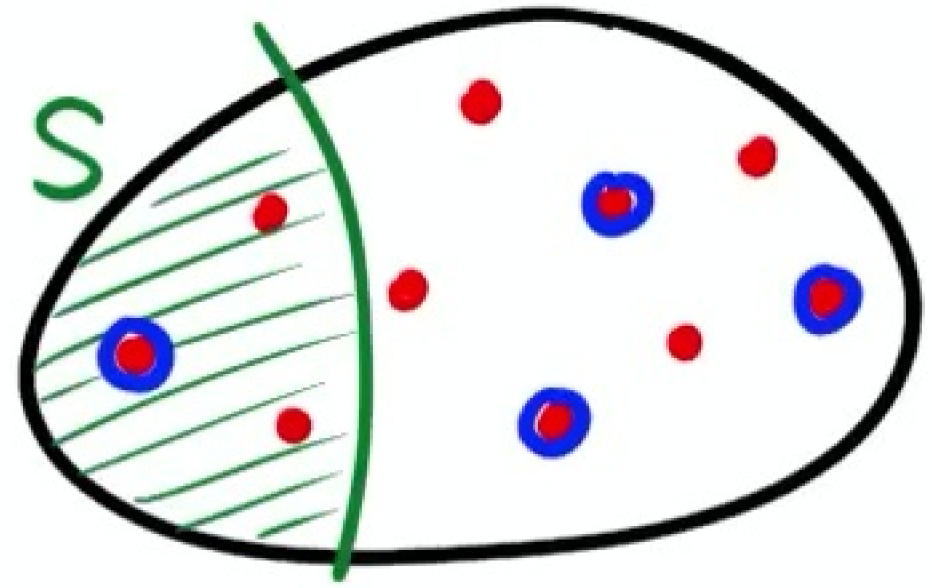
Trivial: |R| s-t mincuts

[L.-Panigrahi '20] O(log |R|) s-t mincuts suffice!

Warm-up: global Steiner mincut (separate at least two terminals)

Sample at rate 1/2, 1/4, 1/8, ...

If sample at rate ~ $\frac{1}{|S \cap R|}$, then constant prob. success



Cut Threshold Algorithm

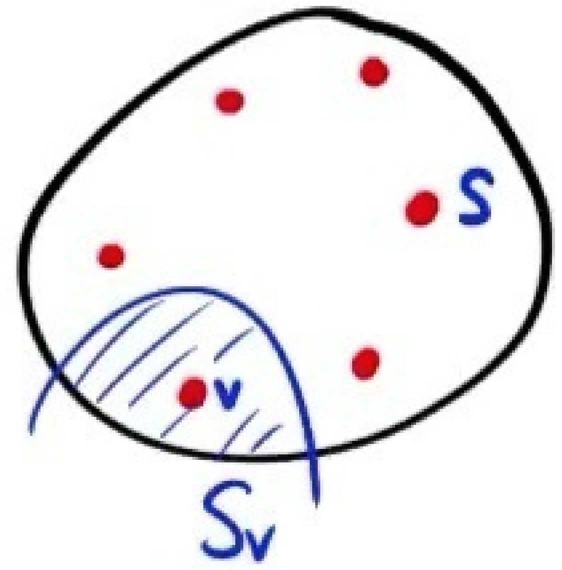
Given s and λ ("cut threshold"), output all $v \in V$ with $\text{mincut}(s,v) \leq \lambda$

Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with mincut(s,v) $\leq \lambda$

Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified mincut(s,u) $\leq \lambda$ for all $u \in S_v$

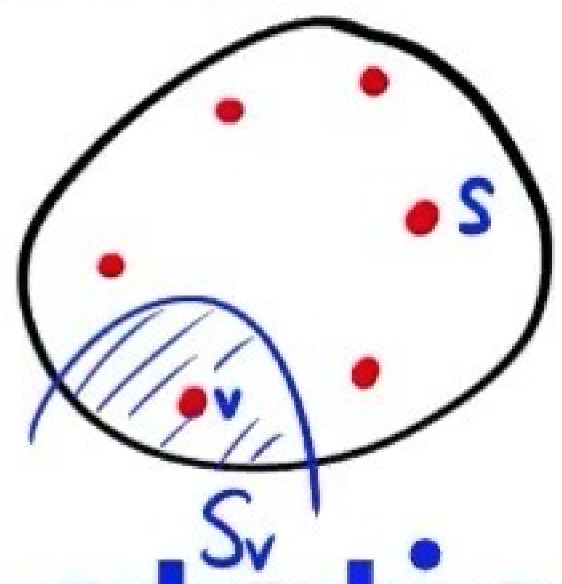


Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with mincut(s,v) $\leq \lambda$

Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified mincut(s,u) $\leq \lambda$ for all $u \in S_v$



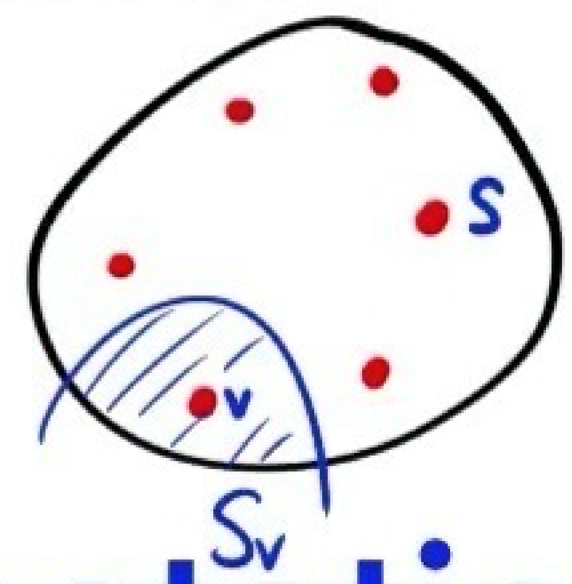
Claim: Certify Ω(1/logn) fraction in expectation

Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with $mincut(s,v) \leq \lambda$

Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified $mincut(s,u) \leq \lambda$ for all $u \in S_v$



Claim: Certify $\Omega(1/\log n)$ fraction in expectation

Algorithm CutThreshold:

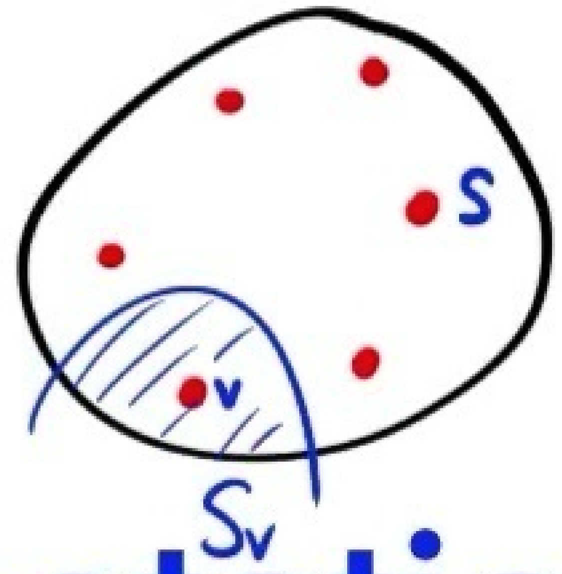
repeat $O(\log^2 n)$ times: call CutThresholdStep(G,s,R)
where R = {still uncertified}

Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with $\text{mincut}(s,v) \leq \lambda$

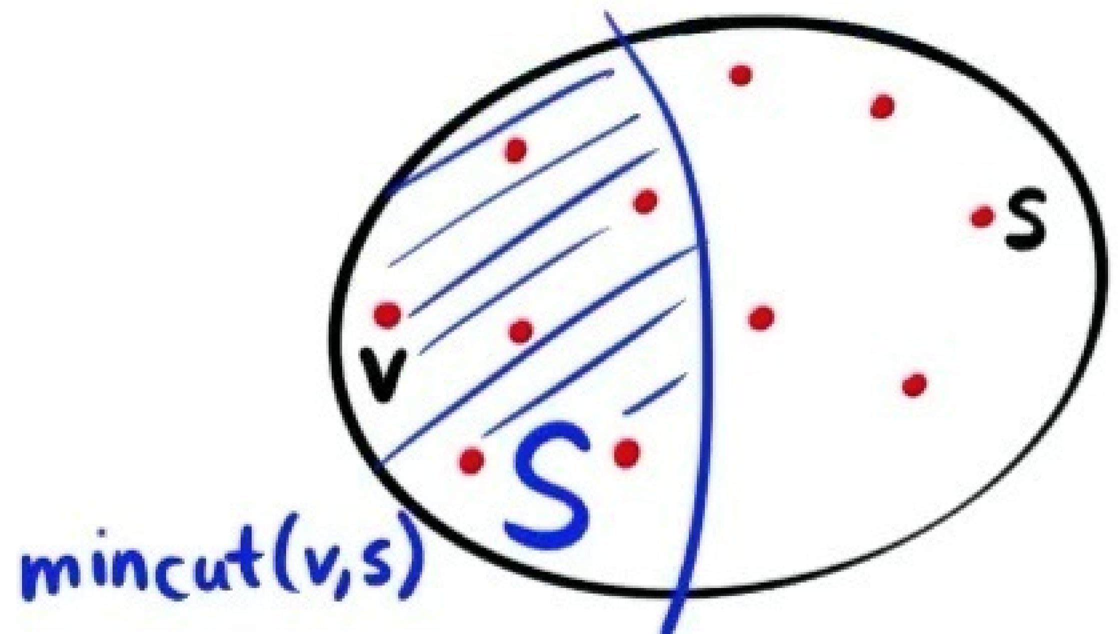
Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified $\text{mincut}(s,u) \leq \lambda$ for all $u \in S_v$



Claim: Certify $\Omega(1/\log n)$ fraction in expectation

Intuition: fix $v \in R$ certifiable ($\text{mincut}(s,v) \leq \lambda$)

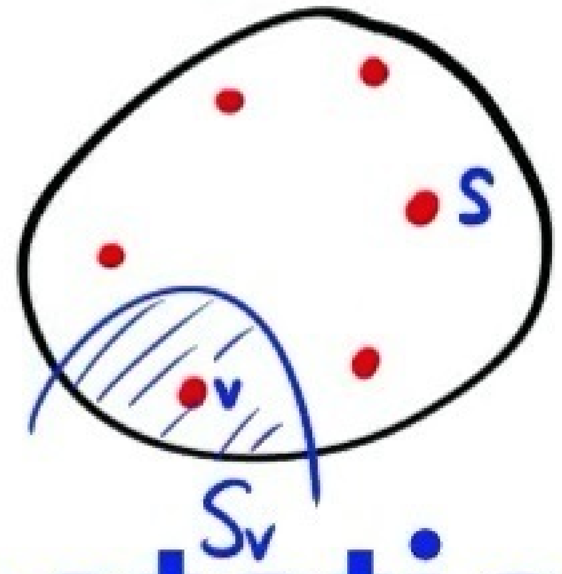


Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with $\text{mincut}(s,v) \leq \lambda$

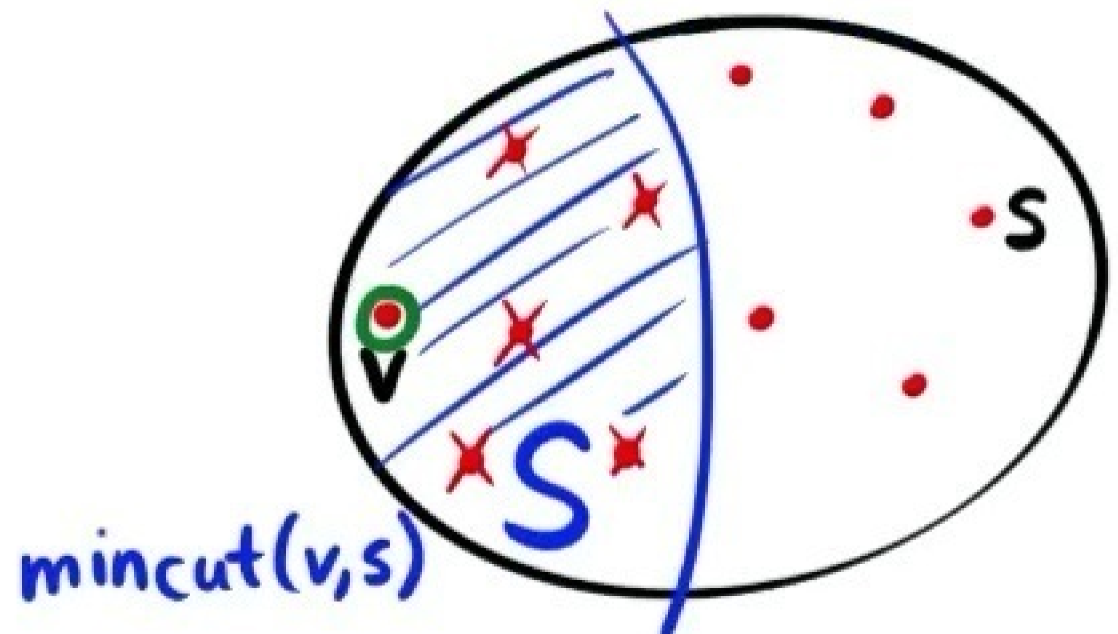
Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified $\text{mincut}(s,u) \leq \lambda$ for all $u \in S_v$



Claim: Certify $\Omega(1/\log n)$ fraction in expectation

Intuition: fix $v \in R$ certifiable ($\text{mincut}(s,v) \leq \lambda$)

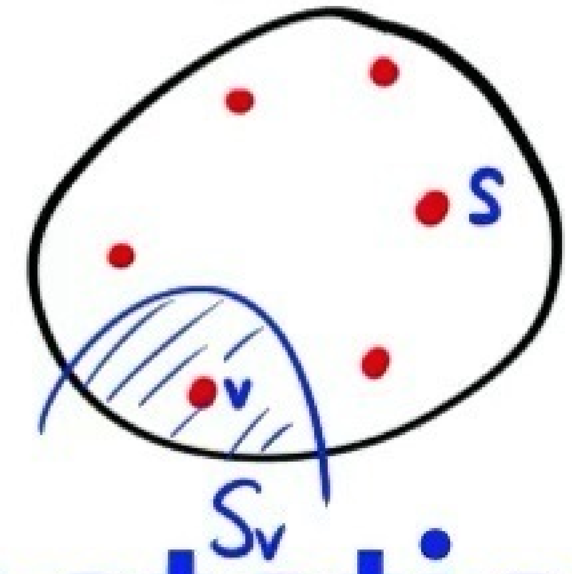


Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with mincut(s,v) $\leq \lambda$

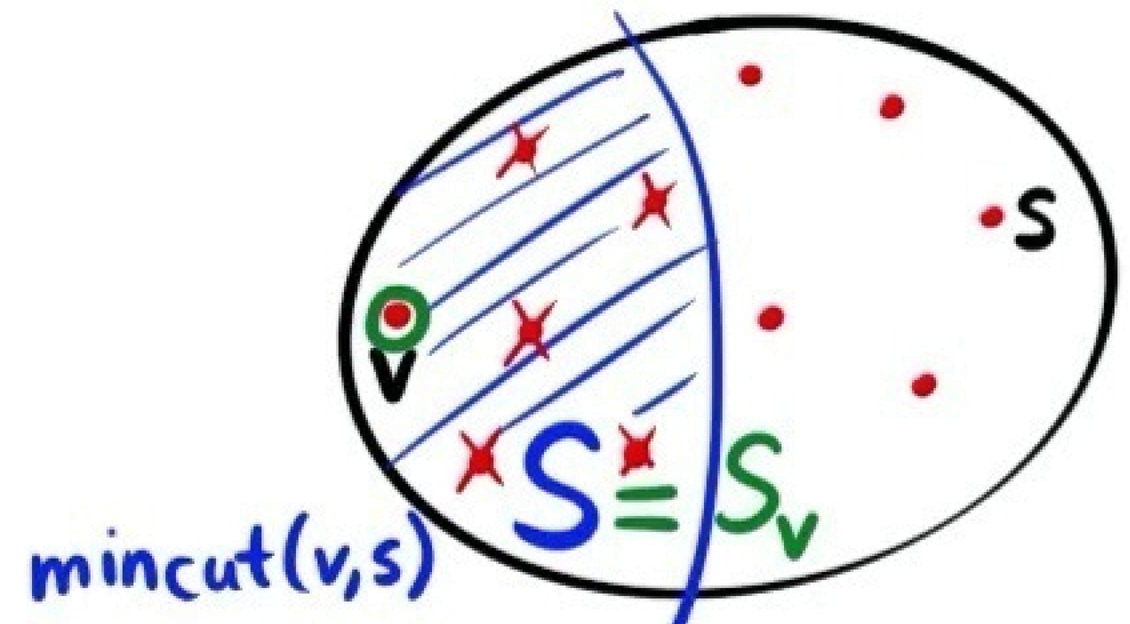
Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified mincut(s,u) $\leq \lambda$ for all $u \in S_v$



Claim: Certify $\Omega(1/\log n)$ fraction in expectation

Intuition: fix $v \in R$ certifiable (mincut(s,v) $\leq \lambda$)

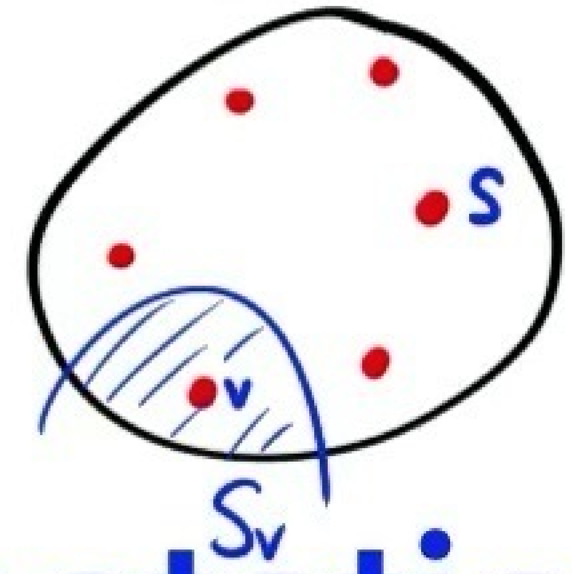


Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with mincut(s,v) $\leq \lambda$

Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified mincut(s,u) $\leq \lambda$ for all $u \in S_v$

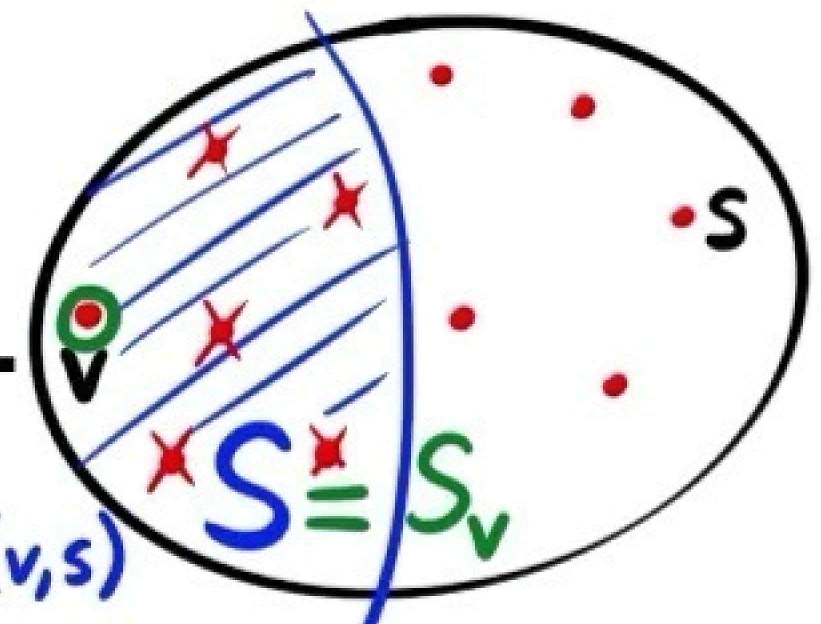


Claim: Certify $\Omega(1/\log n)$ fraction in expectation

Intuition: fix $v \in R$ certifiable (mincut(s,v) $\leq \lambda$)

with prob. ~$1/|S \cap R|$, certify $|S \cap R|$ vertices => E[#certify] $\approx$ 1

over all v certifiable, E[#certify] $\approx$ #certifiable

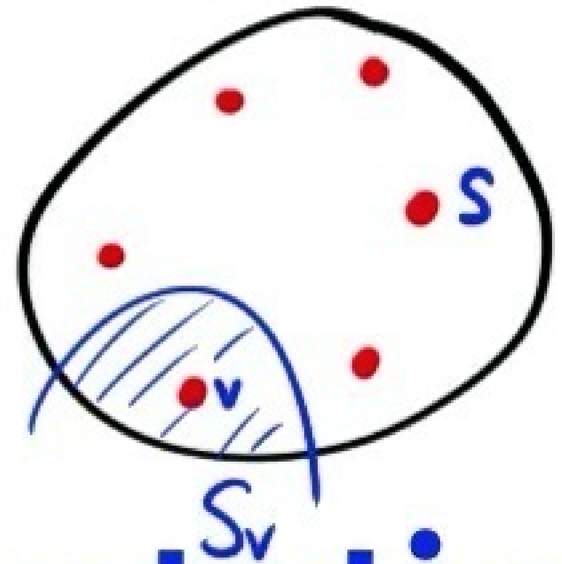


Cut Threshold Algorithm

Given s and λ ("cut threshold"), output all $v \in R$ with mincut(s,v) $\leq \lambda$

Algorithm CutThresholdStep(G, s, R):

- For each i=1,2,...,logn
  - Sample in R with rate $p=1/2^i$ except s is always sampled
  - Compute Minimum Isolating Cuts with sampled set R^i
  - For each set $S_v : v \in R^i \setminus s$,
    if $|\partial S_v| \leq \lambda$ then certified mincut(s,u) $\leq \lambda$ for all $u \in S_v$

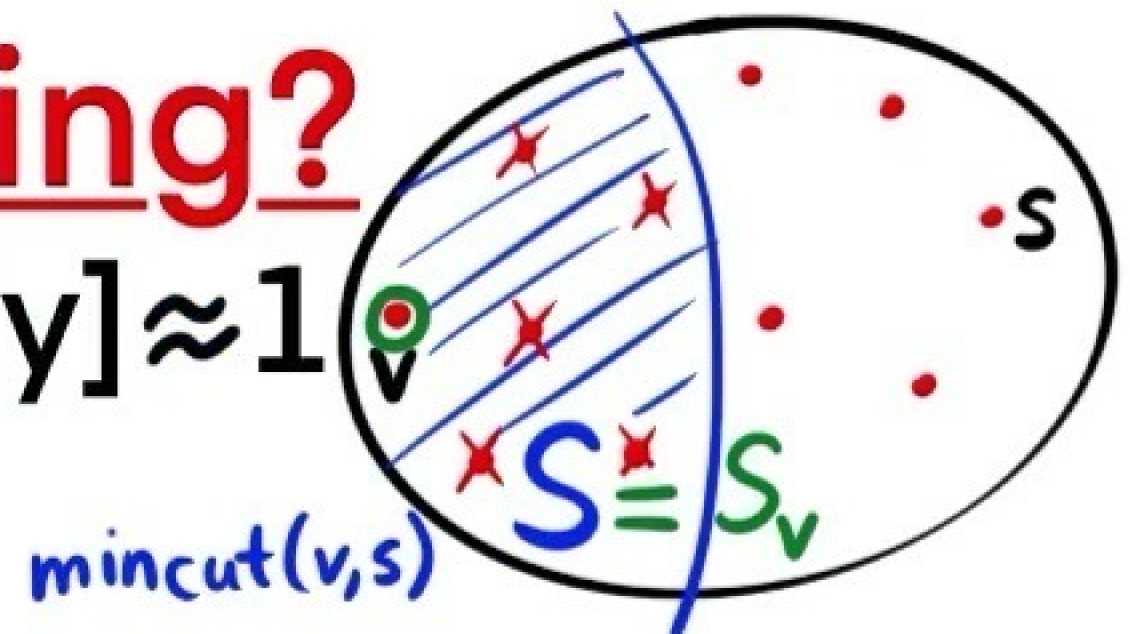


Claim: Certify $\Omega(1/\log n)$ fraction in expectation

Intuition: fix $v \in R$ certifiable (mincut(s,v) $\leq \lambda$) overcounting?

with prob. ~$1/|S \cap R|$, certify $|S \cap R|$ vertices => E[#certify]≈1

over all v certifiable, E[#certify]≈#certifiable



Cut Threshold Algorithm

Algorithm: for each i, sample ~$1/2^i$ (s always sampled)

compute minimum isolating cuts,

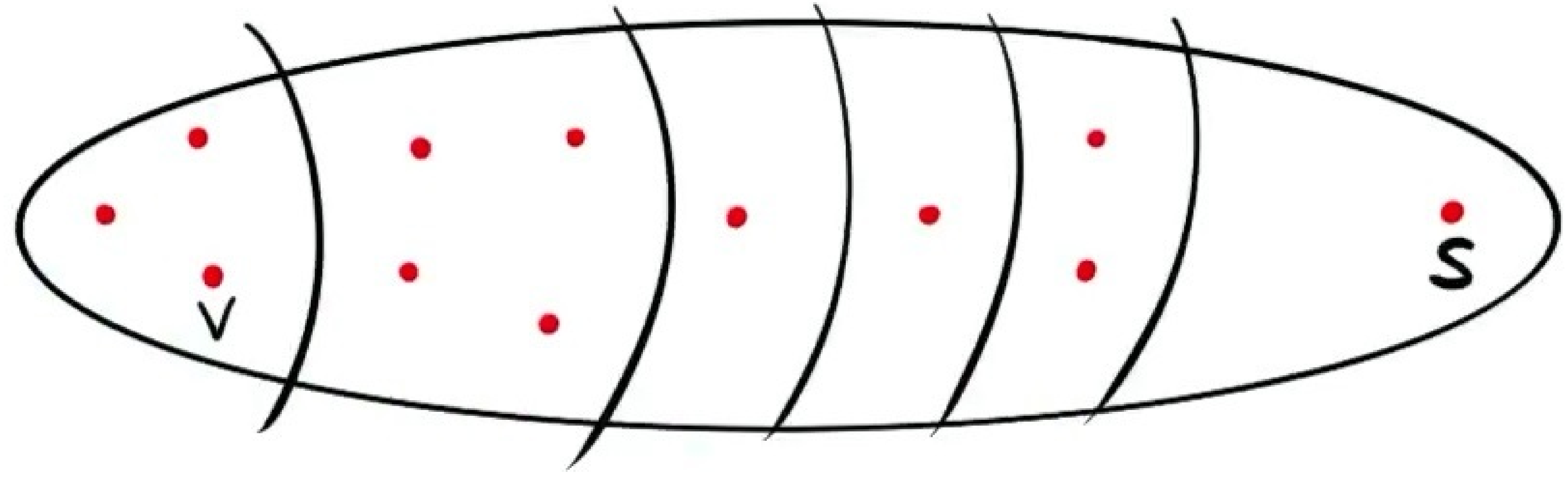
certify all cuts

Cut Threshold Algorithm

Algorithm: for each i, sample ~$1/2^i$ (s always sampled)
compute minimum isolating cuts,
certify all cuts

Claim: each vertex overcounted O(logn) times w.h.p.

Proof: use fact that s-t mincuts are laminar:



Cut Threshold Algorithm

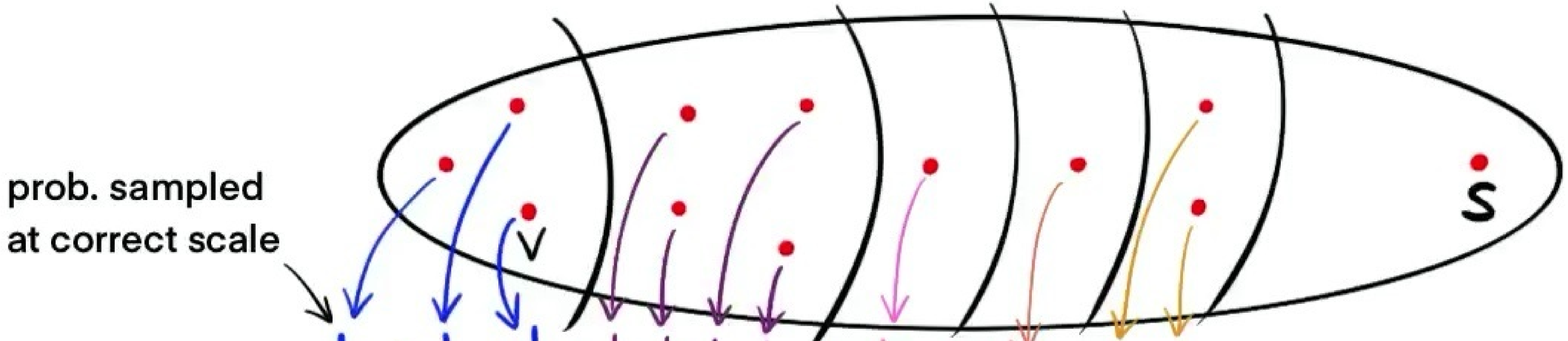
Algorithm: for each i, sample $\sim 1/2^i$ (s always sampled)

compute minimum isolating cuts,

certify all cuts

Claim: each vertex overcounted $O(\log n)$ times w.h.p.

Proof: use fact that s-t mincuts are laminar:



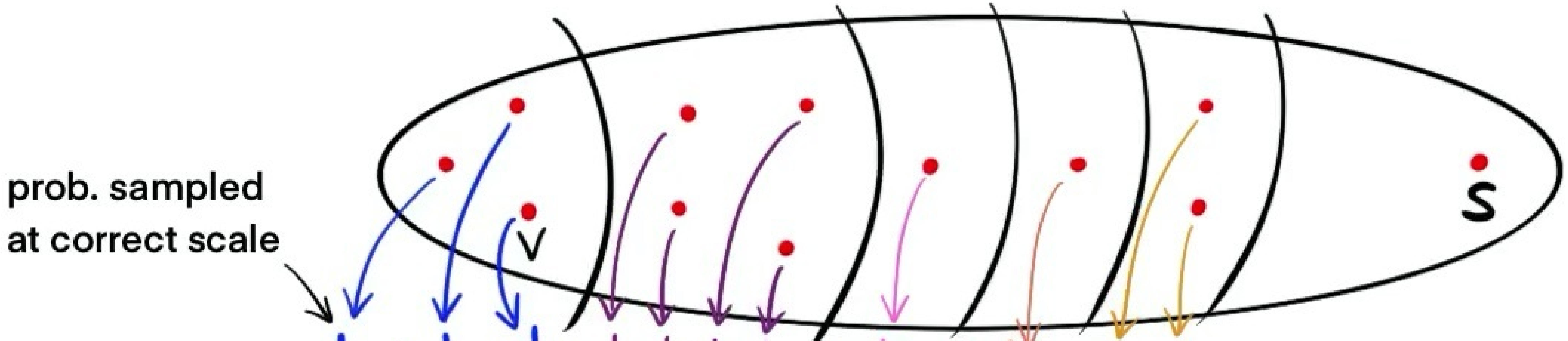
$$E[\#\text{times } v \text{ certified}] \lesssim \frac{1}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{8}+\frac{1}{9}+\frac{1}{11}+\frac{1}{11}$$

Cut Threshold Algorithm

Algorithm: for each i, sample ~$1/2^i$ (s always sampled)
compute minimum isolating cuts,
certify all cuts

Claim: each vertex overcounted O(logn) times w.h.p.

Proof: use fact that s-t mincuts are laminar:



E[#times v certified] $\lesssim \frac{1}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{8}+\frac{1}{9}+\frac{1}{11}+\frac{1}{11}$

Dominated by $\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\cdots+\frac{1}{|R|} \approx \ln|R|$

Cut Threshold Algorithm

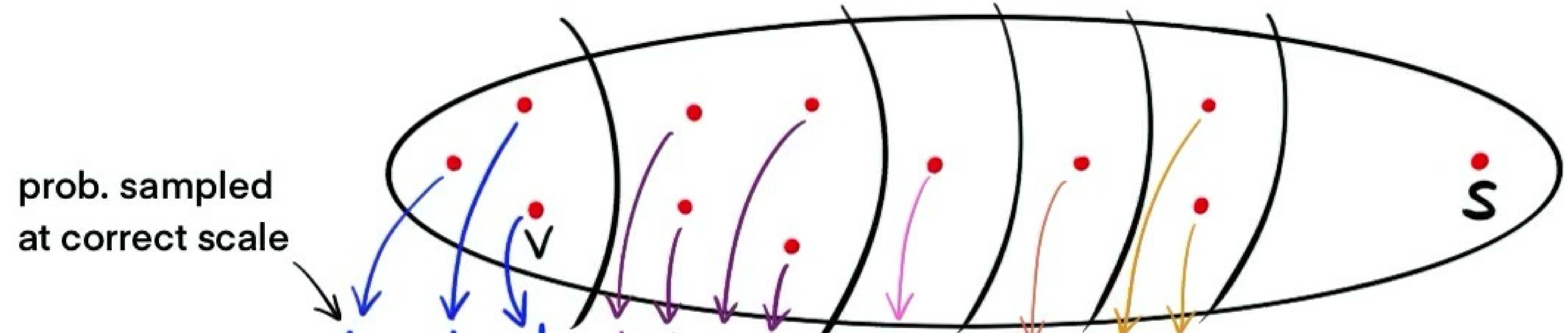
Algorithm: for each i, sample ~$1/2^i$ (s always sampled)

compute minimum isolating cuts,

certify all cuts

Claim: each vertex overcounted O(logn) times w.h.p.

Proof: use fact that s-t mincuts are laminar:



E[#times v certified] $\lesssim \frac{1}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{8}+\frac{1}{9}+\frac{1}{11}+\frac{1}{11}$

Dominated by $\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\cdots+\frac{1}{|R|} \approx \ln|R|$

E[#times v certified] $\lesssim$ O(log |R|). Chernoff bound: O(log n) w.h.p.

Approximate SSMC and GH Tree

SSMC algorithm: run Cut Threshold on each $\lambda=(1+\varepsilon)^i$

Approximate SSMC and GH Tree

SSMC algorithm: run Cut Threshold on each $\lambda = (1+\varepsilon)^i$

Gomory-Hu tree algorithm:

standard recursive approach, except multiple branches

Approximate SSMC and GH Tree

SSMC algorithm: run Cut Threshold on each $\lambda = (1+\varepsilon)^i$

Gomory-Hu tree algorithm:

standard recursive approach, except multiple branches

select s uniformly at random [AKT'20]

Control the approximation factor (nontrivial)

Conclusion and Open Questions

Algorithms for Cut Threshold, $(1+\varepsilon)$-SSMC, $(1+\varepsilon)$-GHTree in ~max-flow time

Main tools: minimum isolating cuts + random sampling

Conclusion and Open Questions

Algorithms for Cut Threshold, $(1+\varepsilon)$-SSMC, $(1+\varepsilon)$-GHTree in ~max-flow time

Main tools: minimum isolating cuts + random sampling

Open questions:

- Faster exact GH tree? Reduces to exact SSMC values
- SSMC values faster than n-1 max-flows?
- Approximate GH tree in near-linear time?
- More applications of Minimum Isolating Cuts?