Graph Connectivity Using Star Contraction

(logs Matter)

Sagnik Mukhopadhyay

UNIVERSITY OF BIRMINGHAM

Joint work with Simon, Troy, Yuval, Pawel and Danupon. Appeared in FOCS 2022.

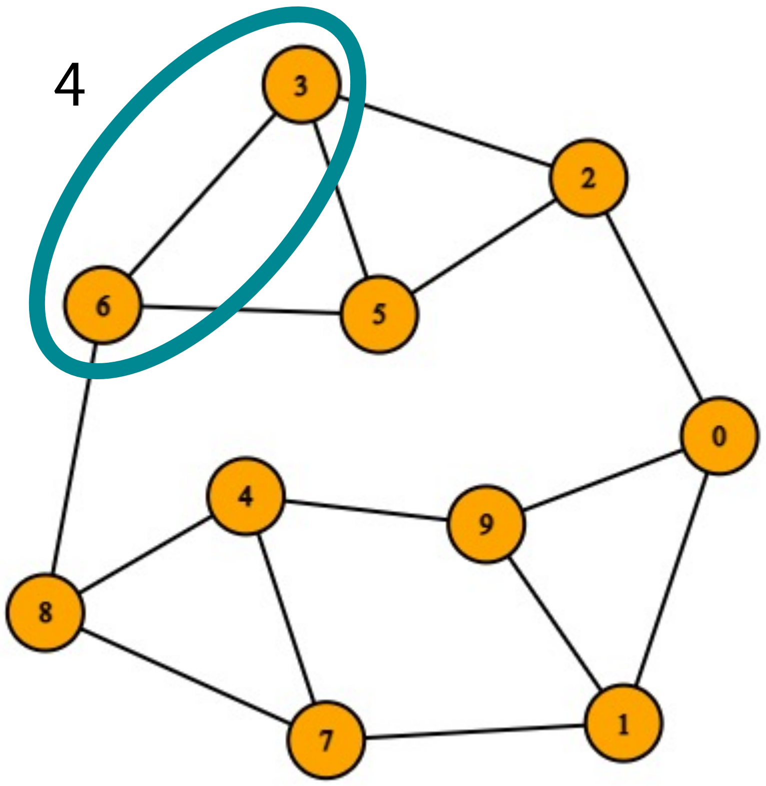
Model of Computation

Cut queries – Min Cut

- Given $G = (V, E)$, access via *cut queries:*

$$S \subseteq V \Rightarrow |E(S, V \setminus S)|$$

- **Goal**: find a minimum cut, denoted C.
- δ – minimum degree
- λ – edge connectivity

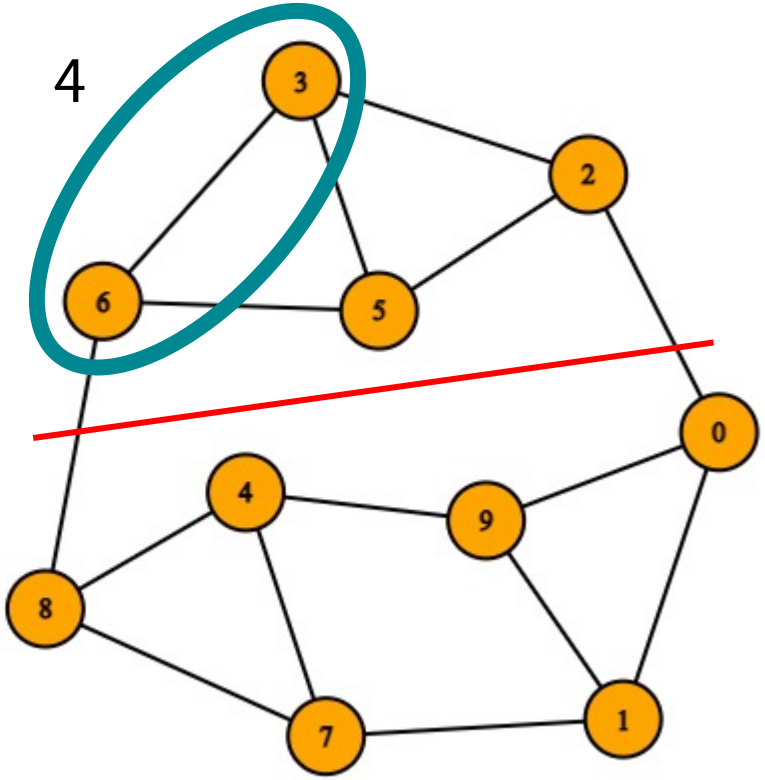


Cut queries – Min Cut

- Given $G = (V, E)$, access via *cut queries:*

$$S \subseteq V \Rightarrow |E(S, V \backslash S)|$$

- **Goal**: find a minimum cut, denoted C.

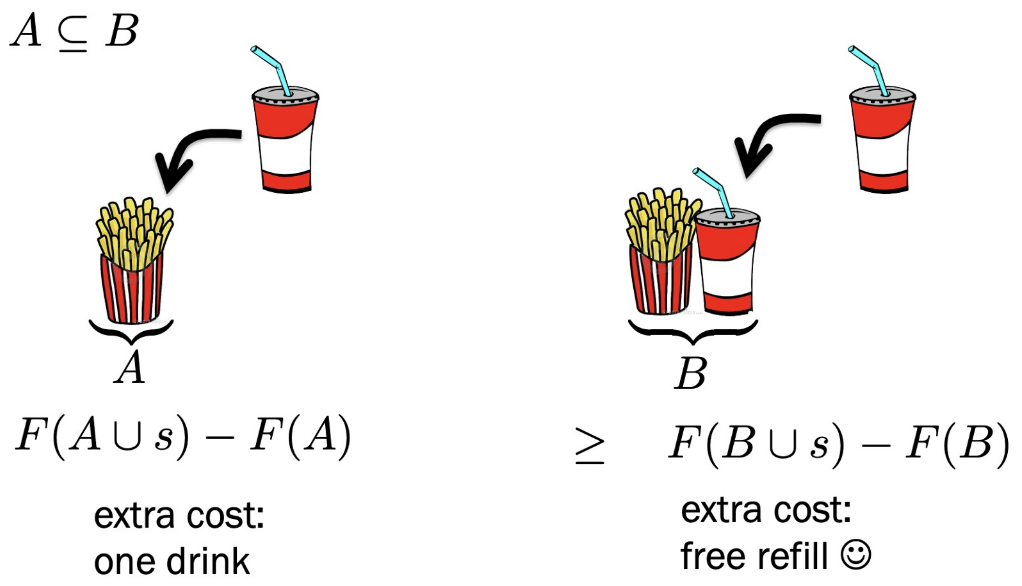


Trivial: $O(n^2)$, learn the graph. $|E(S,T)|$ in $O(1)$ queries.

Motivation – Submodular function minimization

- $F: 2^V \rightarrow \mathbb{R}$ is sub-modular if $\forall S, T \in 2^V, F(S) + F(T) \geq F(S \cup T) + F(S \cap T)$
- Query access.
- Goal: find $\arg\min_{S \in 2^V} F(S)$.
- Examples:
  - **Graph cuts, $\mathbf{F}(S) = |\partial S|$**
  - Entropy
  - Mutual Information
  - Matroid rank

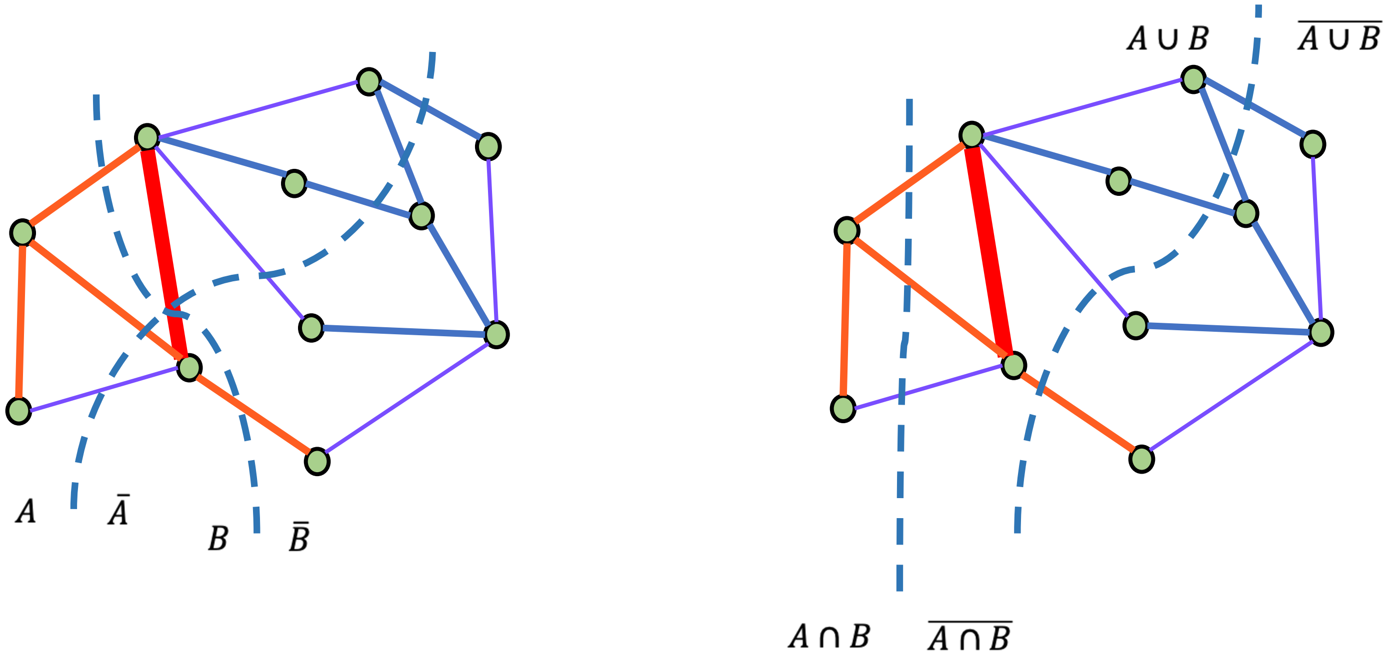
Diminishing marginal gain



Slides inspired by and figures taken from https://people.csail.mit.edu/stefje/mlss/kyoto_mlss_lecture1.pdf

Motivation – Submodular function minimization

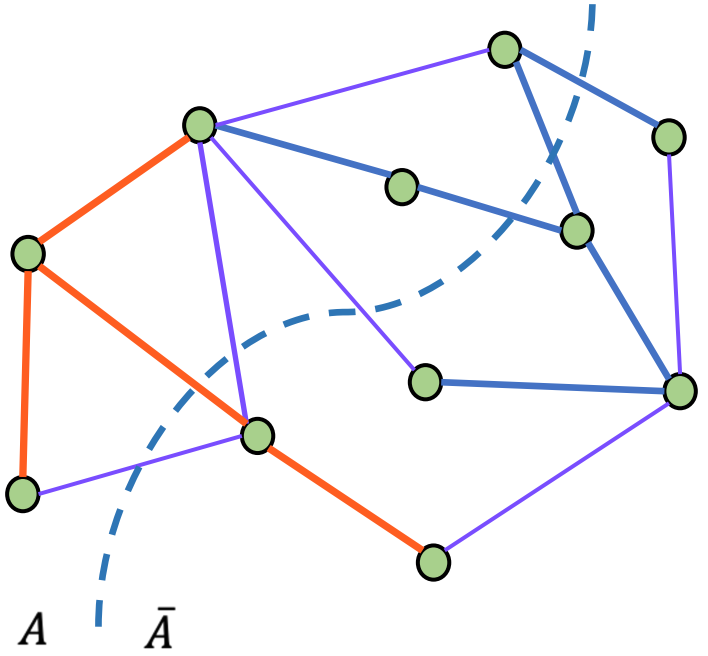
- $F: 2^V \to \mathbb{R}$ is sub-modular if $\forall S, T \in 2^V, F(S) + F(T) \geq F(S \cup T) + F(S \cap T)$
- Query access.
- Goal: find $\arg\min_{S \in 2^V} F(S)$.



$$\text{Cut}(A) + \text{Cut}(B) > \text{Cut}(A \cap B) + \text{Cut}(A \cup B)$$

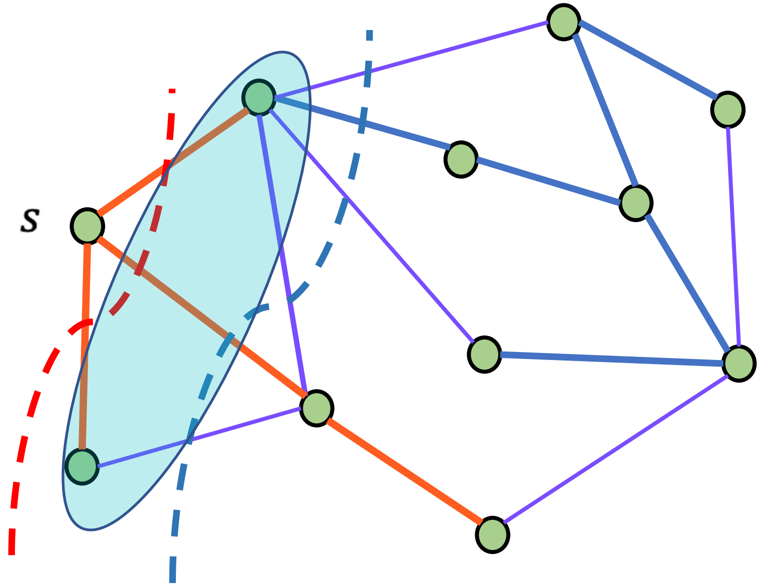
7 3 3 5

Motivation – Symm Submodular function minimization



$$\mathrm{Cut}(A) = \mathrm{Cut}(V - A)$$

Global Min-cut: Goal is **non-trivial minimizer**



$$\mathrm{Cut}(A) \neq \mathrm{Cut}(V - A)$$

(s, t)-Min-cut = Max-Flow ⇐ **Bipartite matching**

SFM – Previous work, upper bounds

	Upper bound	Det/Ran	Combinatorial?	SymSFM/SFM
Grotschel, Lovasz, Schrijver, 1988	$\tilde{O}(n^5)$	Det	No	SFM
Iwata, Fleischer, Fujishige 2001	$\tilde{O}(n^7)$	Det	Yes	SFM
Iwata, Orlin 2009	$\tilde{O}(n^5)$	Det	Yes	SFM
Jiang 2021	$O(n^2 \log n)$	Det	No	SFM

SFM – Previous work, upper bounds

	Upper bound	Det/Ran	Combinatorial?	SymSFM/SFM
Grotschel, Lovasz, Schrijver, 1988	$\tilde{O}(n^5)$	Det	No	SFM
Iwata, Fleischer, Fujishige 2001	$\tilde{O}(n^7)$	Det	Yes	SFM
Iwata, Orlin 2009	$\tilde{O}(n^5)$	Det	Yes	SFM
Jiang 2021	$O(n^2 \log n)$	Det	No	SFM
Jiang 2021+[**MN**, CQ21]	$\tilde{O}(n^2)$	Ran	No*	symSFM
Queyranne 1998	$O(n^3)$	Det	Yes	symSFM

SFM – Previous work, Lower bounds

	Lower bound	Det/Ran	Applies to min cut?	SymSFM/SFM
Hajnal, Mass, Turán 1988, Harvey 2008	$\Omega(n)$	Det	Yes	symSFM
Babai, Frankl, Simon 1986	$\Omega\left(\frac{n}{\log n}\right)$	Ran	Yes	symSFM
Chakrabarty, Graur, Jiang, Sidford 2022	$\Omega(n \log n)$	Det	No	SFM

Lower bound situation is dire!

What problems are suitable for proving high SFM lower bound?

Previous Work

- **Connectivity** in $O(n \log n)$ cut queries [Harvey 2008]
- Unweighted minimum cut in $O(n \log^3 n)$ cut queries [Rubinstein, Schramm, Weinberg 2018]
- **Multigraph** minimum cut in $O(n \log^4 n)$ cut queries [M, Nanongkai 2020]

} Combinatorial!

- $\Omega\left(\frac{n}{\log n}\right)$ cut queries for **Connectivity**, $\Omega(n)$ assuming communication complexity conjecture of [Babai, Frankl, Simon 1986]
- $\Omega\left(\frac{n \log \log n}{\log n}\right)$ cut queries for minimum cut on simple graphs [Assadi, Dudeja 2021]

Results

Main Result

Theorem. Randomised cut-query algorithm for min-cut in simple graphs has $O(n)$ complexity.

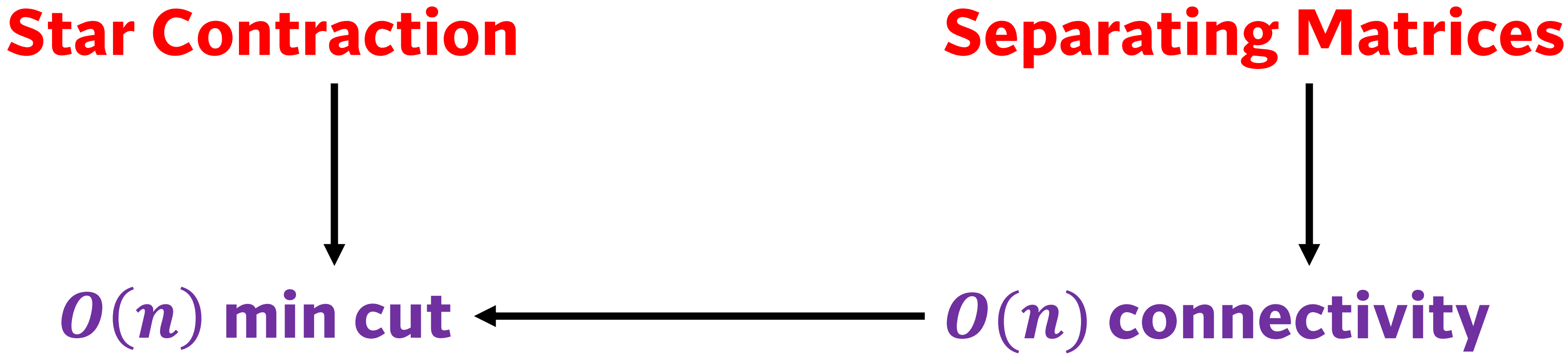
Improves state of the art even for connectivity!

Tight under conjecture of **[Babai, Frankl, Simon 1986]**

Other applications: Matrix-vector queries, semi streaming etc.

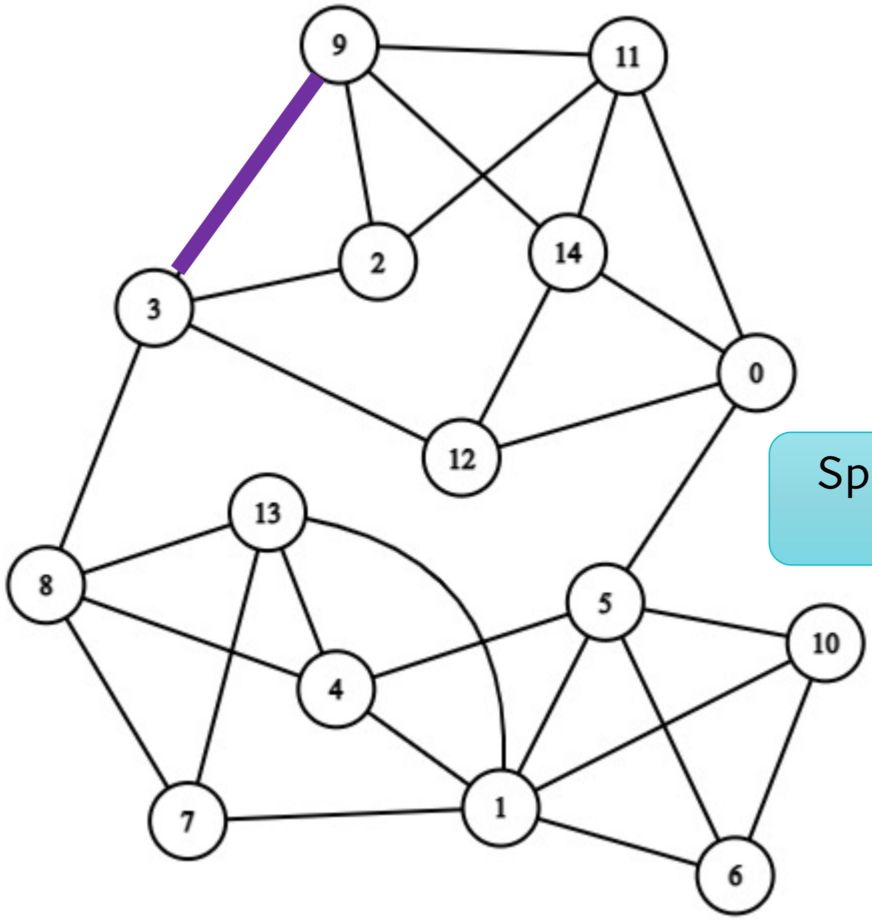
Main Result

Theorem. Randomised cut-query algorithm for min-cut in simple graphs has $O(n)$ complexity.



Techniques

Background: Cut Query Primitives

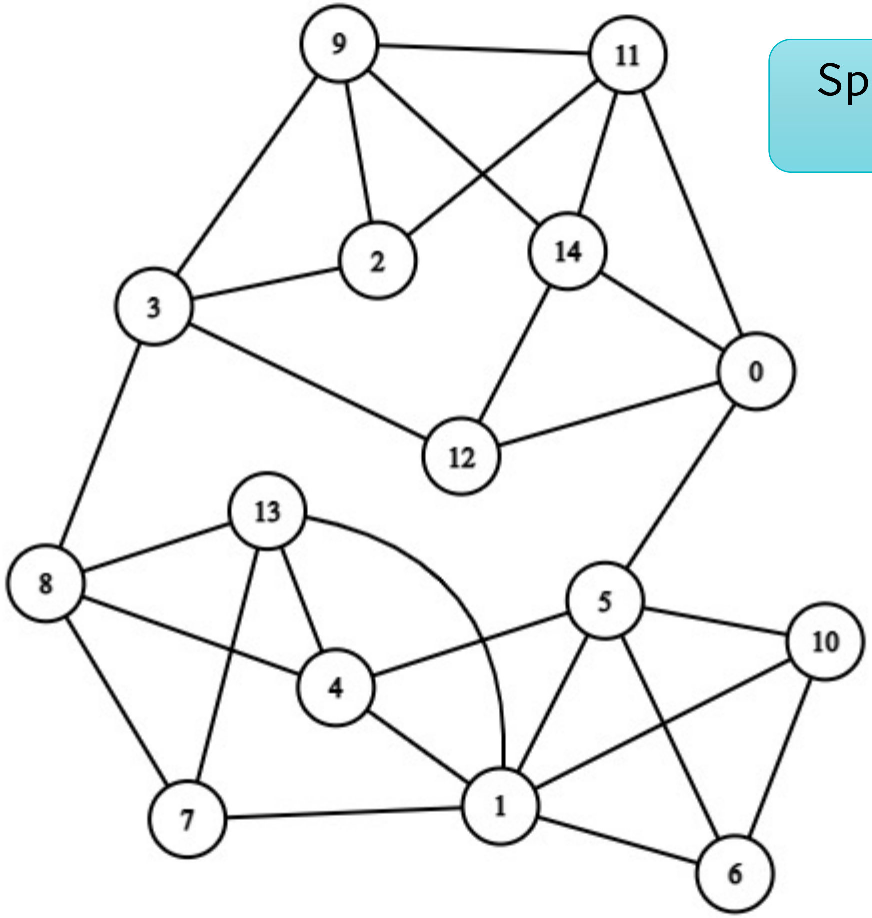


- Can check if an edge (u, v) is present
  - 3 queries
- Can find an edge between v and S
  - $O(\log n)$ queries

Spanning tree: $\tilde{O}(n)$ queries

- Can randomly sample an edge
  - Incident on v: $O(\log n)$ queries
  - Any: $O(n \log n)$ queries
- Can work on graph minors.*

Background: Basic Algorithm



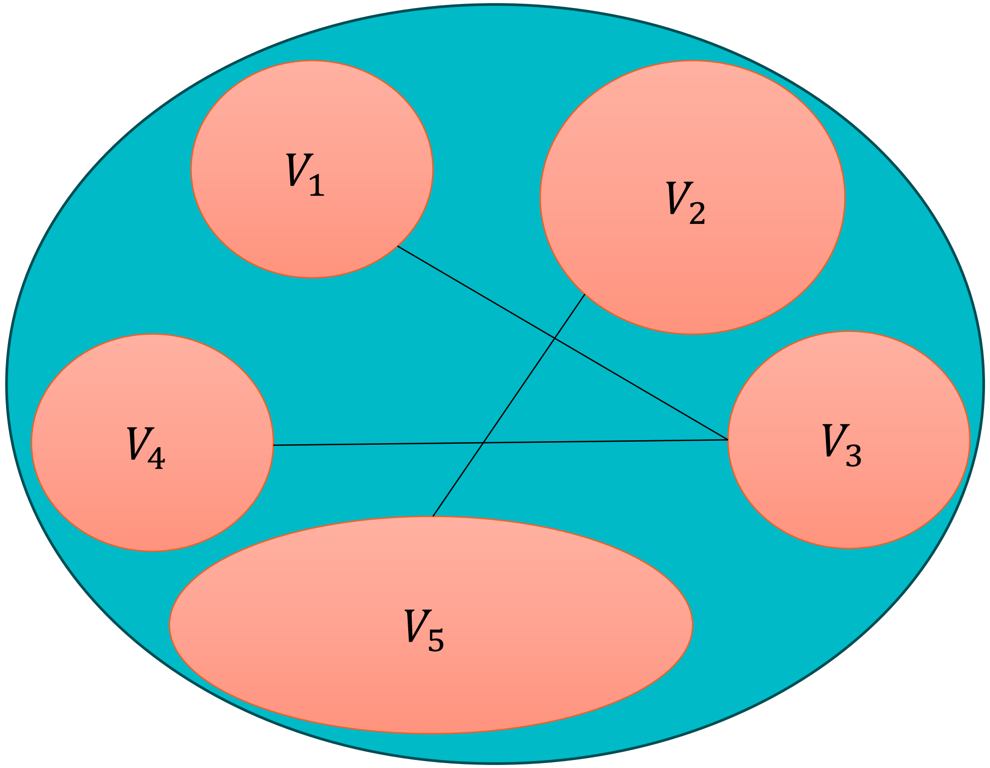
Spanning tree: $\tilde{O}(n)$ queries

- Pack δ spanning trees.
  - Each tree must cross every cut at least once.
  - $\delta \geq \lambda$.
- Complexity: $\tilde{O}(n\delta) \rightarrow O(n\delta)$.

Separating matrices

- Can we do any better?

Background: Min-cut Preserving Clustering [Kawarabayashi, Thorup 2015]



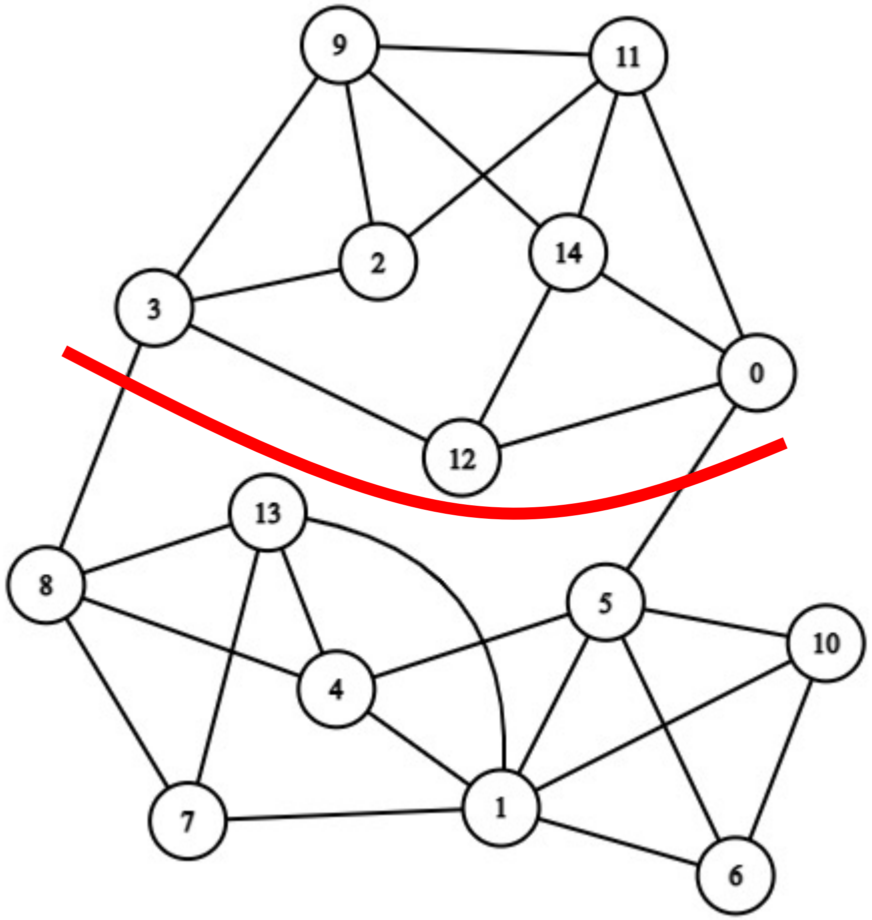
$G = (V, E)$

- Simple graph G with min deg δ.
- Contract: $G \rightarrow G'$ such that
  - G' has $\tilde{O}\left(\frac{n}{\delta}\right)$ vertices and $\tilde{O}(n)$ edges.
  - All non-trivial min-cuts are preserved.

Min-cut (G) = Min-cut (G')

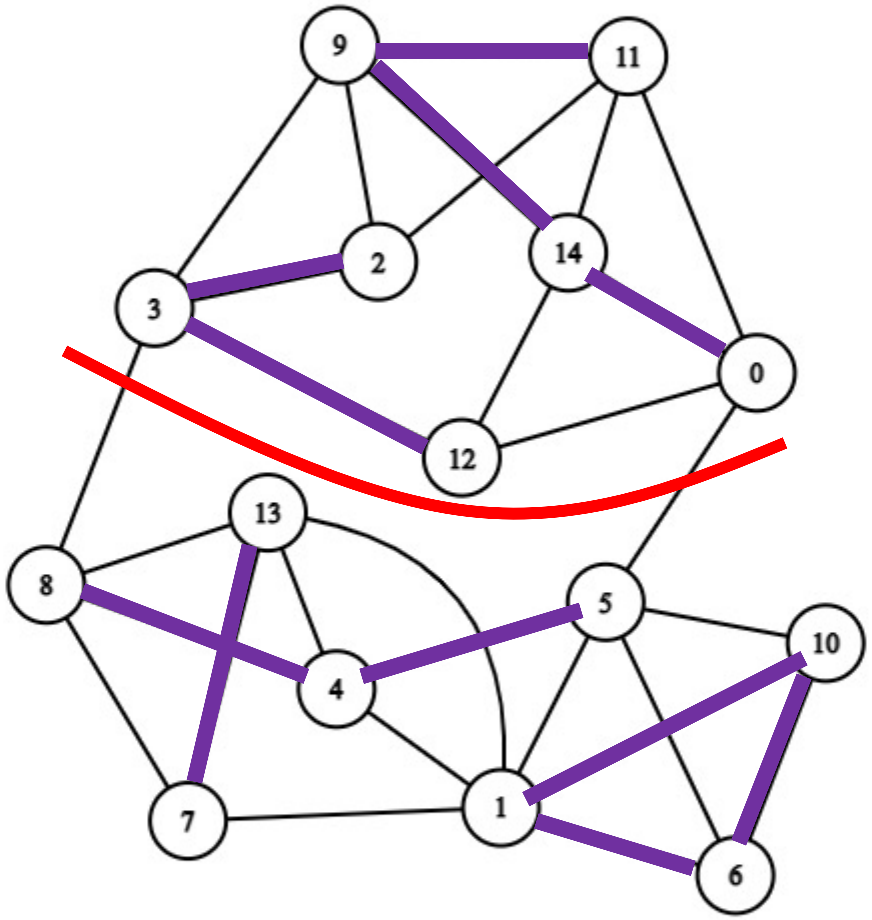
- Pack δ spanning trees in G'
  - Linear complexity

Background: One-out contraction



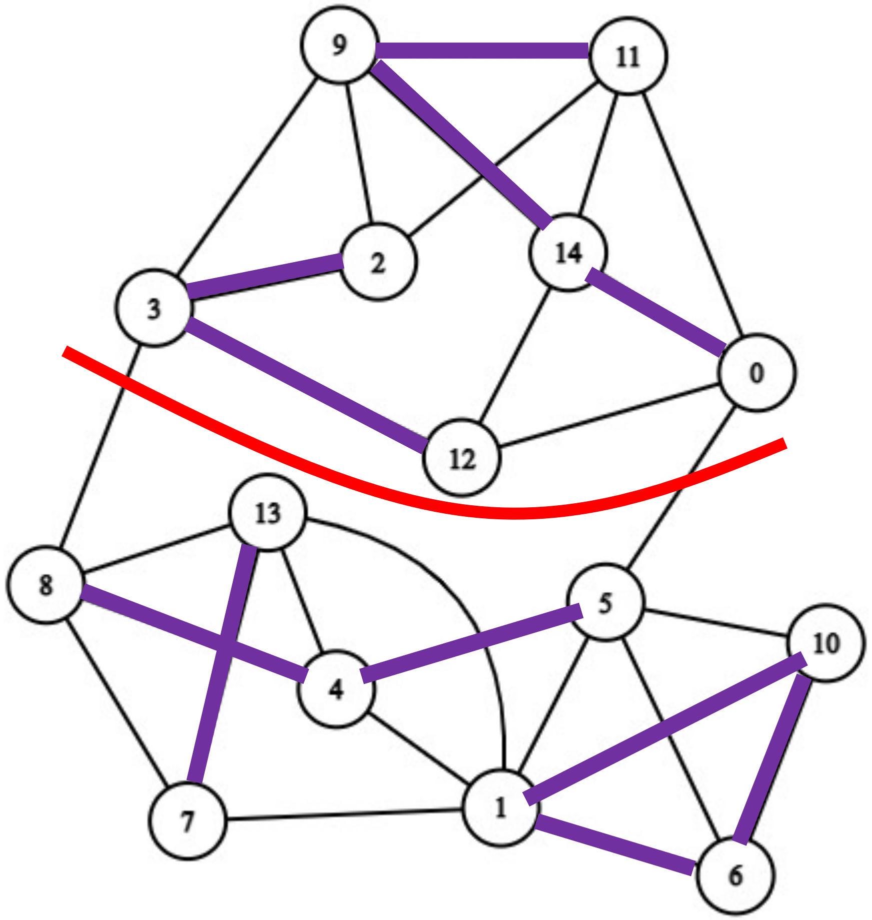
- Let C be some min cut. Every vertex v chooses uniformly random neighbor $u \in N(v)$.

Background: One-out contraction



- Let C be some min cut. Every vertex v chooses uniformly random neighbor $u \in N(v)$.
- **S- sampled edges.**
- $\Pr[S \cap C = \emptyset]$?
- $\Pr[S \cap C = \emptyset] \geq \frac{1}{2}^4 = \frac{1}{16}$
- Constant Prob!

Background: One-out contraction



- Let C be some min cut. Every vertex v chooses uniformly random neighbor $u \in N(v)$.
- ***S*- sampled edge**
- $\Pr[S \cap C = \emptyset] = \prod_{v \in N(C)} \left(1 - \frac{c(v)}{d(v)}\right)$ $\geq \frac{1}{16}$

$$\frac{c(v)}{d(v)} \leq 1/2 \text{ for every } v \in N(C)$$

$$\sum_{v \in N(C)} \frac{c(v)}{d(v)} \leq 2\frac{|C|}{\delta(G)} \leq 2.$$

Background: One-out contraction

$$G = (V, E)$$

↓ 1-out sample + contract edges

$$G' = (V', E')$$

- With constant probability, $\lambda(G) = \lambda(G')$
- Solve on multigraph G'?
- One can show: Exists graphs s.t. $|V'| = \Theta\left(\frac{n}{\sqrt{\delta}}\right)$ w.h.p.

We want: $O\left(\frac{n}{\delta}\right)$

Background: **Two**-out contraction [Ghaffari, Nowicki, Thorup 2020]

$G = (V, E)$

↓ **2**-out sample + contract edges

$G' = (V', E')$

Room for improvement:
- Cluster diameter $\Theta(\log^2 n)$
- Analysis made simpler

2-out sample complexity: $O(n \log n)$.

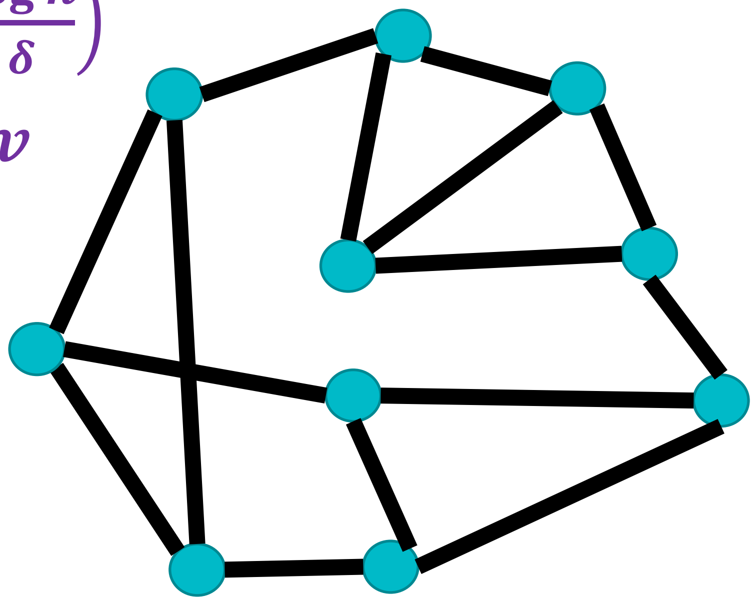
- With constant probability, $\lambda(G) = \lambda(G')$
- Solve on multigraph G'.
- Non-trivial analysis: $|V'| = O\left(\frac{n}{\delta}\right)$ w.h.p.

No logs!!

Complexity: $O\left(\frac{n}{\delta} \cdot \delta\right) = O(n)$.

Star-Contraction

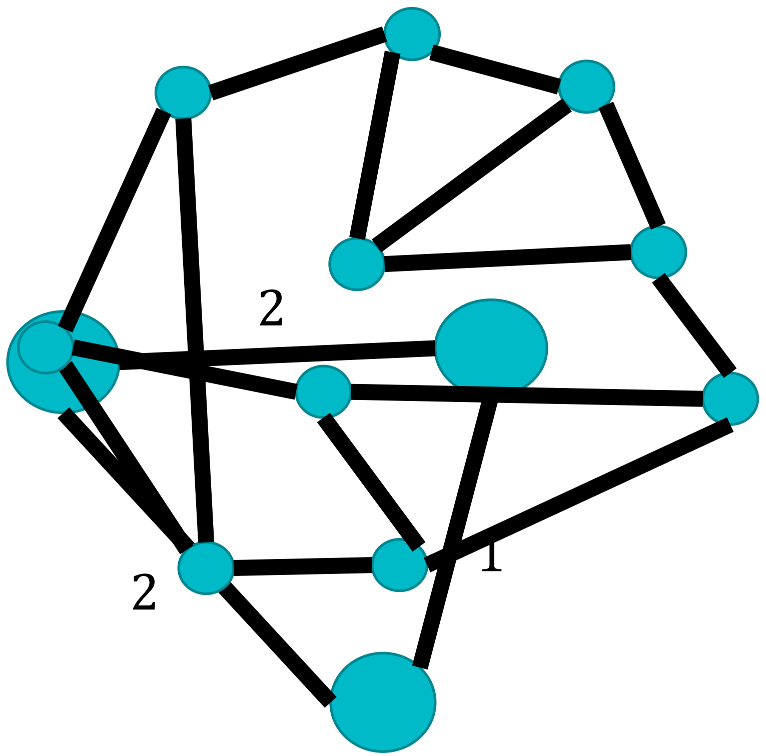
- Idea: Sample from a subset of neighbors
- **Construct set R with each $v \in R$ w.p. $p = \Theta\left(\frac{\log n}{\delta}\right)$**
- **Each $u \notin R$ independently samples neighbor $v \in N_R(u)$**
- Contract sampled edges $\boldsymbol{S}$ into $G' = (V', E')$.



$$\mathbb{E}_R\left[\frac{c_R(v)}{d_R(v)} \,\middle|\, d_R(v) > 0\right] = \frac{c(v)}{d(v)} \quad .$$

Star-Contraction

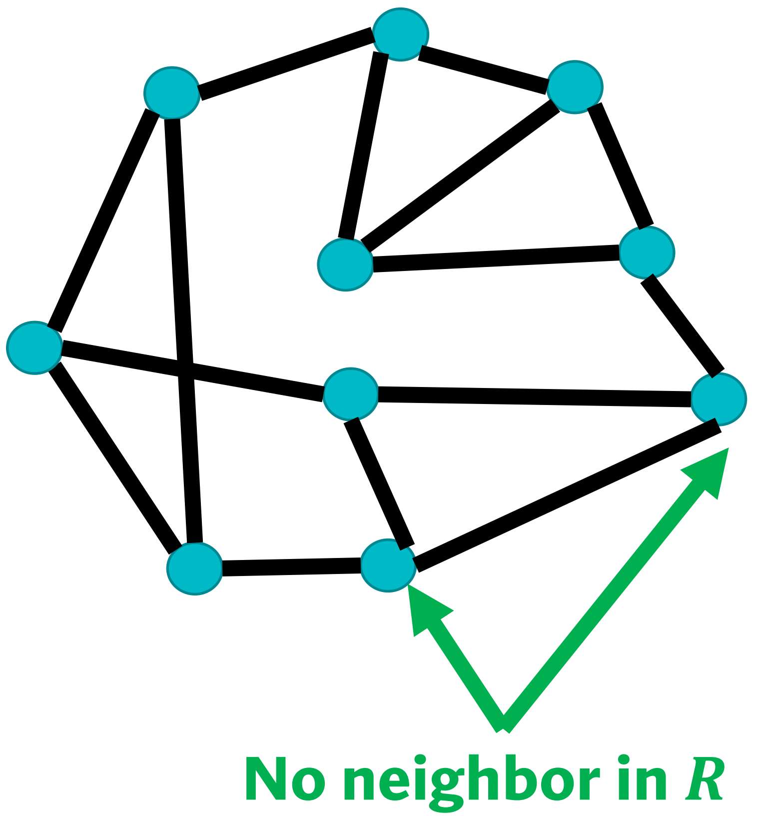
- Idea: Sample from a subset of neighbors
- **Construct set R with each $v \in R$ w.p. $p = \Theta\left(\frac{\log n}{\delta}\right)$**
- **Each $u \notin R$ independently samples neighbor $v \in N_R(u)$**
- Contract sampled edges S into $G' = (V', E')$.
- **Immediate**: $|V'| = O\left(\frac{n \log n}{\delta}\right)$ w.h.p.
- $\Pr[S \cap C = \emptyset] = \Omega(1)$.
- $\lambda(G') = \lambda(G)$ with constant probability!



Complexity: $O\left(\frac{n}{\delta}\log n \,.\, \delta\right) = O(n \log n)$.

Beyond $n \log n$ step 1: Refined star contraction

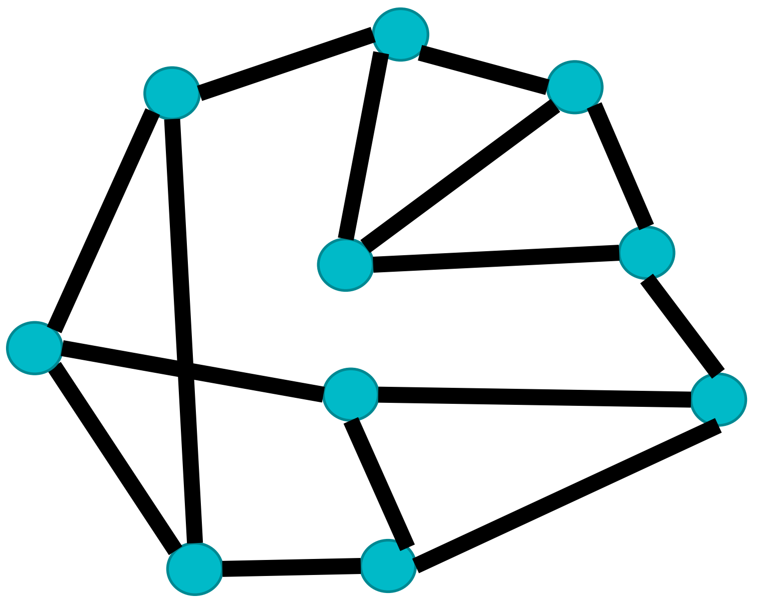
- $|R|$ and therefore $|V'|$ are too large
- Replace $p = \Theta\left(\frac{\log n}{\delta}\right)$ with $p = \Theta\left(\frac{\log \delta}{\delta}\right)$
- **Immediate**: $|R| = O\left(\frac{n \log \delta}{\delta}\right)$
- $|V^*| = |\{v \in V \mid N(v) \cap R = \emptyset| = O\left(\frac{n}{\delta}\right)$
- $V' = R \cup V^*$
- Each $u \notin V'$, independently samples neighbor $v \in N_R(u)$
- Contract into $G' = (V', E')$
- $O(n \log \delta) \Rightarrow O(n \log \log n)$ queries.



If $\delta \geq polylog(n)$, use **[Mukhopadhyay, Nanongkai 2020]**

Beyond $n \log n$ step 1: Refined star contraction

- $|R|$ and therefore $|V'|$ are too large
- Replace $p = \Theta\left(\frac{\log n}{\delta}\right)$ with $p = \Theta\left(\frac{\log \delta}{\delta}\right)$
- **Immediate**: $|R| = O\left(\frac{n \log \delta}{\delta}\right)$
- $|V^*| = |\{v \in V \mid N(v) \cap R = \emptyset| = O\left(\frac{n}{\delta}\right)$
- $V' = R \cup V^*$
- Each $u \notin V'$, independently samples neighbor $v \in N_R(u)$
- Contract into $G' = (V', E')$
- $O(n \log \delta) \Rightarrow O(n \log \log n)$ queries.



How?

Trivial: $O(n \log n)$

Separating Matrices

If $\delta \geq polylog(n)$, use **[Mukhopadhyay, Nanongkai 2020]**

The Problem

Each $u \notin V'$ independently samples neighbor $v \in N_R(u)$

$$\Pr[S \cap C = \emptyset] = \prod_{v \in N(C)} \left(1 - \frac{c(v)}{d(v)}\right) > \frac{1}{16} \text{ as long as}$$

$$\frac{c(v)}{d(v)} \leq 1/2 \text{ for every } v \in N(C)$$

$$\sum_{v \in N(C)} \frac{c(v)}{d(v)} \leq 2\frac{|C|}{\delta(G)} \leq 2.$$

$$q_u = \Pr_{v:(u,v)\in A}[\{u,v\} \in C]$$

(α, β)-good for contraction if:

1. max property: $\max_u q_u \leq \alpha$, *and*
2. sum property: $\sum_u q_u \leq \beta$.

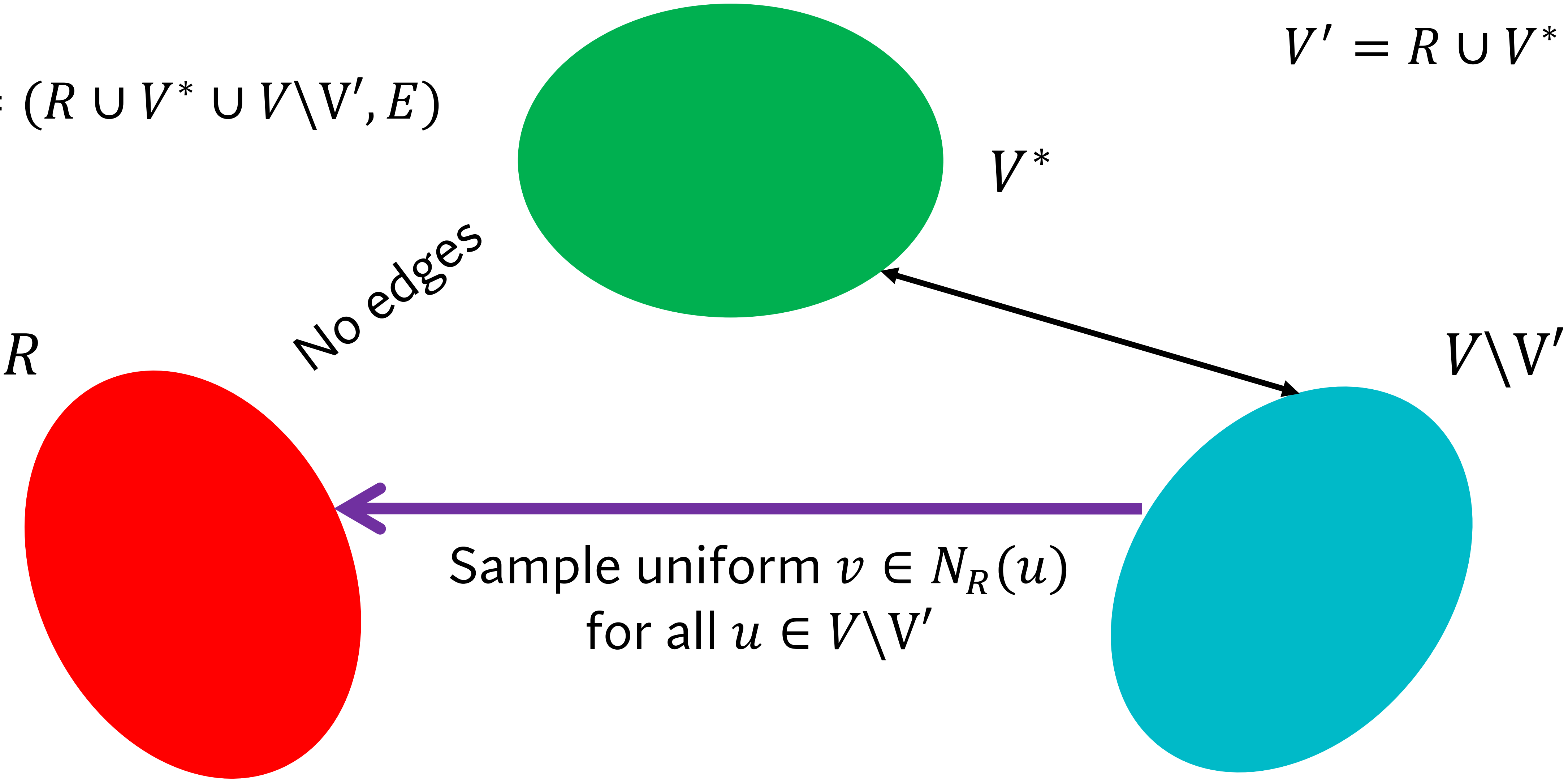
$\Rightarrow$ $\Pr[S \cap C = \emptyset] = (1-\alpha)^{\lceil \beta/\alpha \rceil}.$

The Problem

Each $u \notin V'$ independently samples neighbor $v \in N_R(u)$

$V' = R \cup V^*$

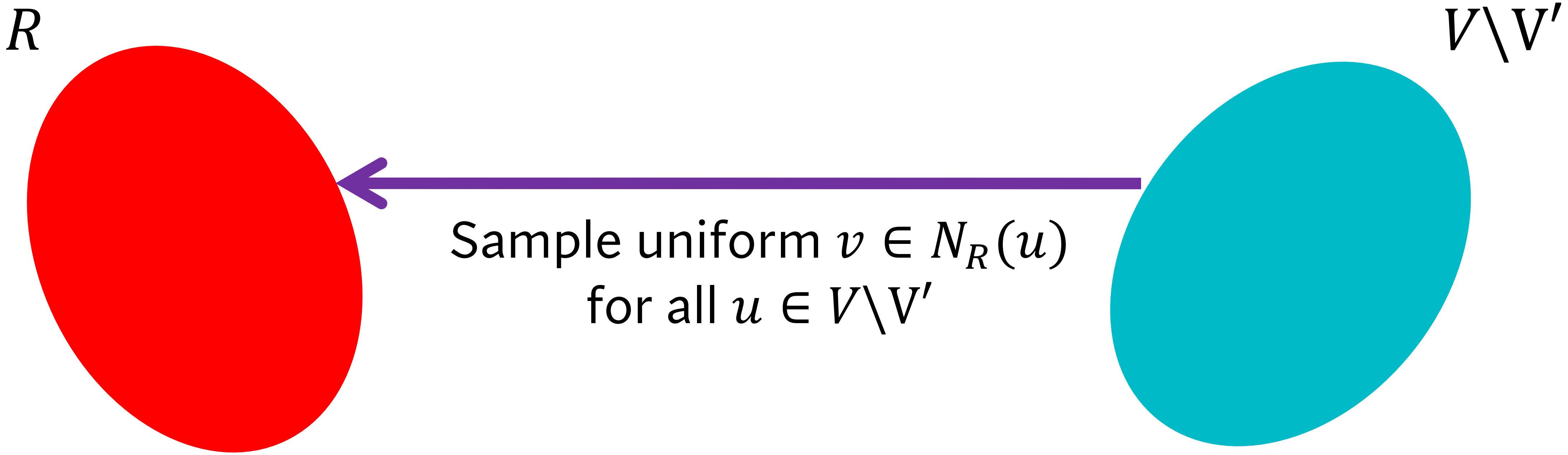
$G = (R \cup V^* \cup V \backslash V', E)$



The Problem

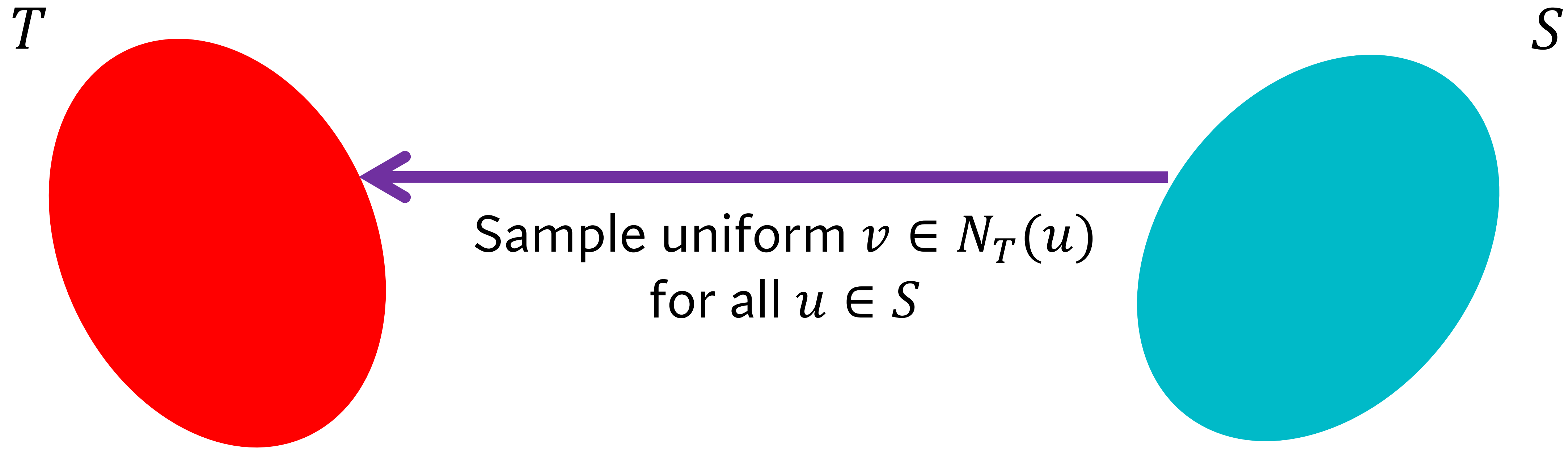
Each $u \notin V'$ independently samples neighbor $v \in N_R(u)$

$V' = R \cup V^*$



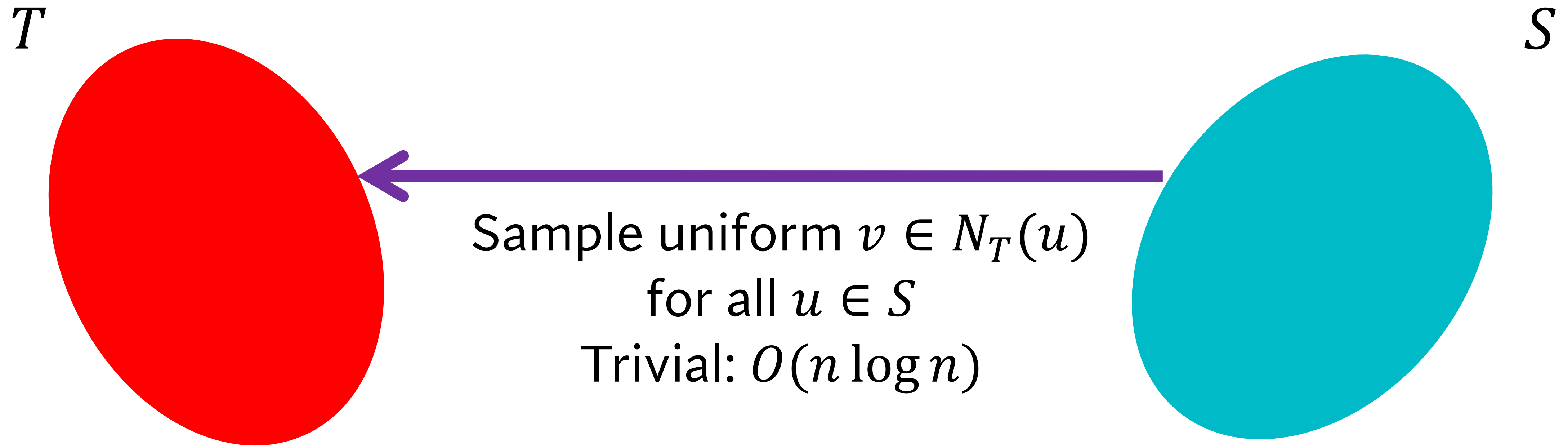
The Problem

Each $u \in S$ independently samples neighbor $v \in T$



The Problem

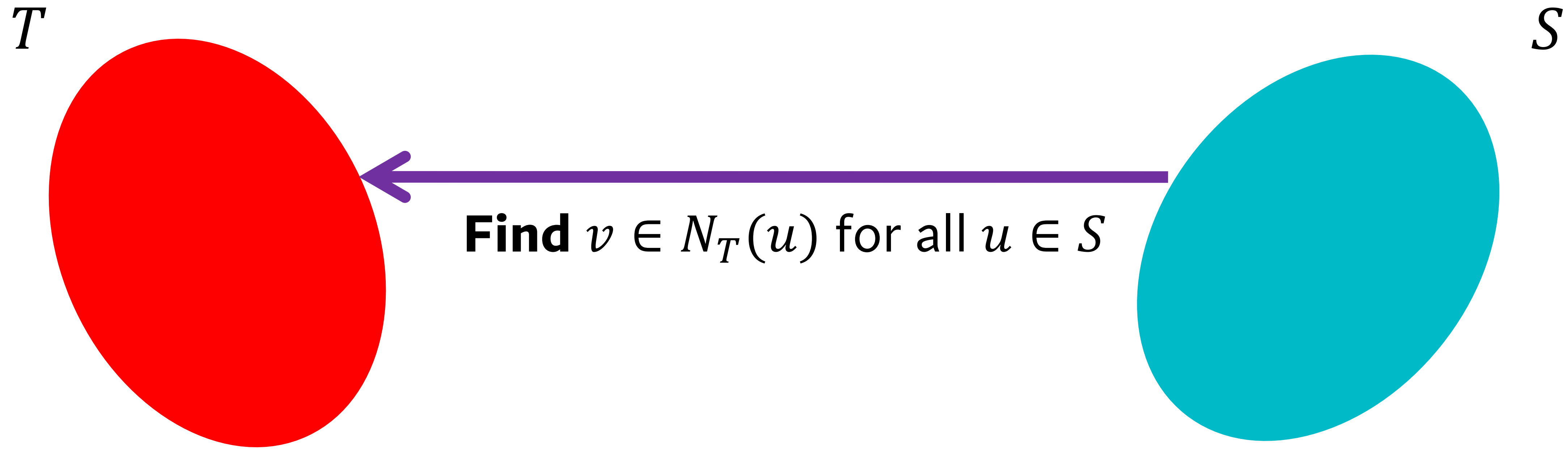
Each $u \in S$ independently samples neighbor $v \in T$



Sample uniform $v \in N_T(u)$
for all $u \in S$
Trivial: $O(n \log n)$

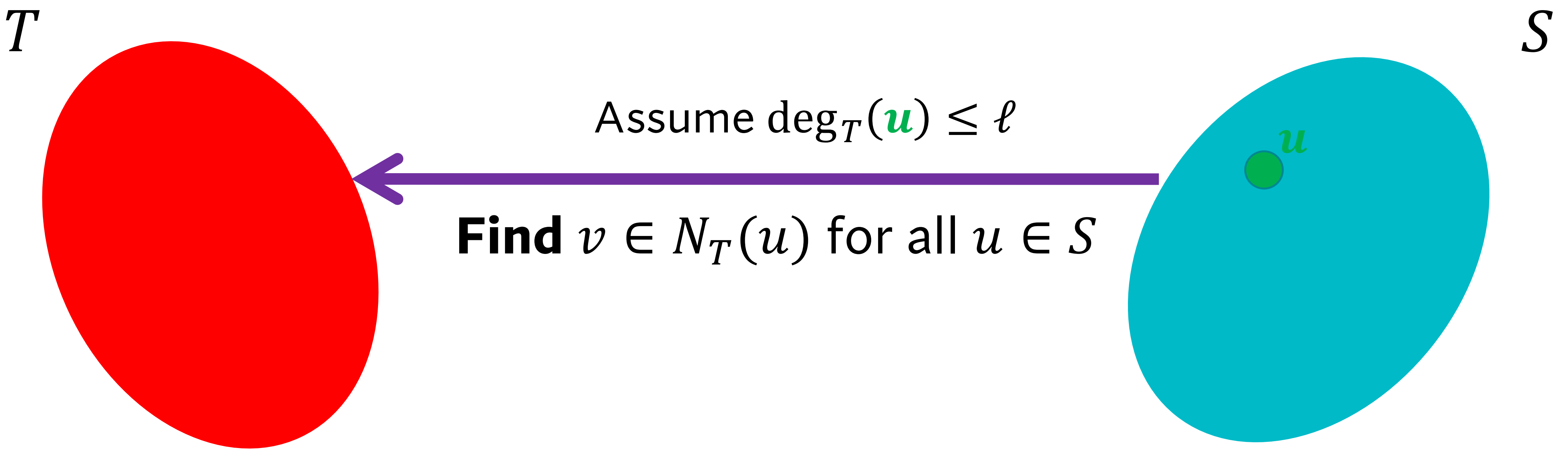
The (Easy) Problem

Each $u \in S$ **finds** a neighbor $v \in T$



The (Easy) Problem

Each $u \in S$ **finds** a neighbor $v \in T$

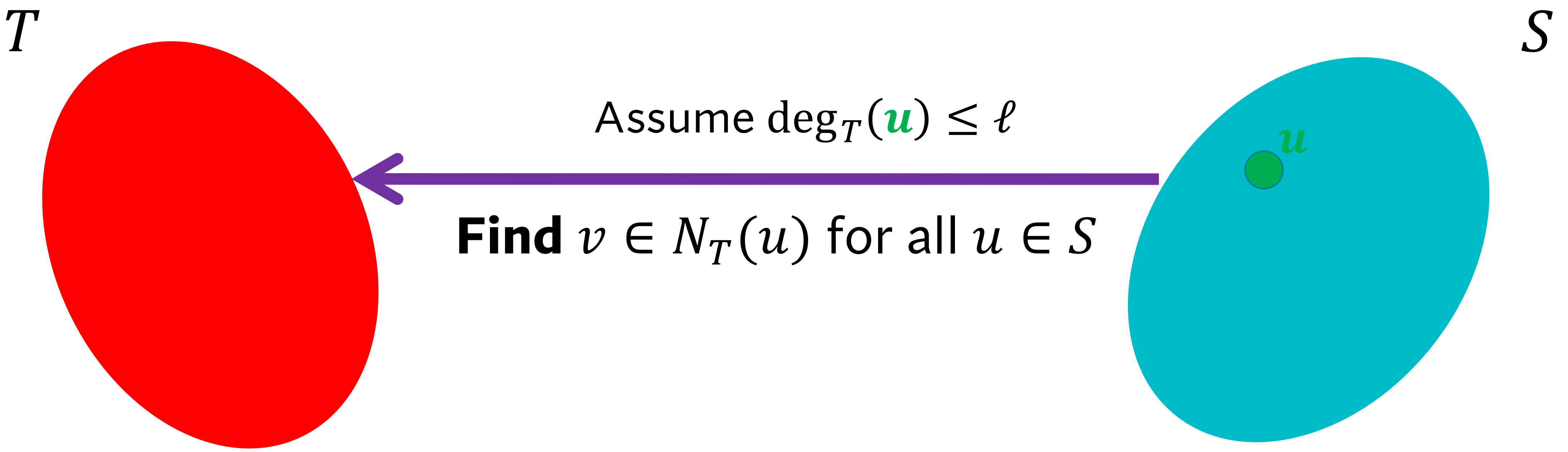


- $|S| \times |T|$ Boolean matrix $M, M[u, v] = 1 \leftrightarrow (u, v) \in E$
- M has row sparsity ℓ
- Goal, learn M!

Seperating Matrices: Learn $M \in \{0,1\}^{n \times n}$ of row sparsity ℓ with $O(\ell n)$ cut queries

The (Easy) Problem

Each $u \in S$ **finds** a neighbor $v \in T$

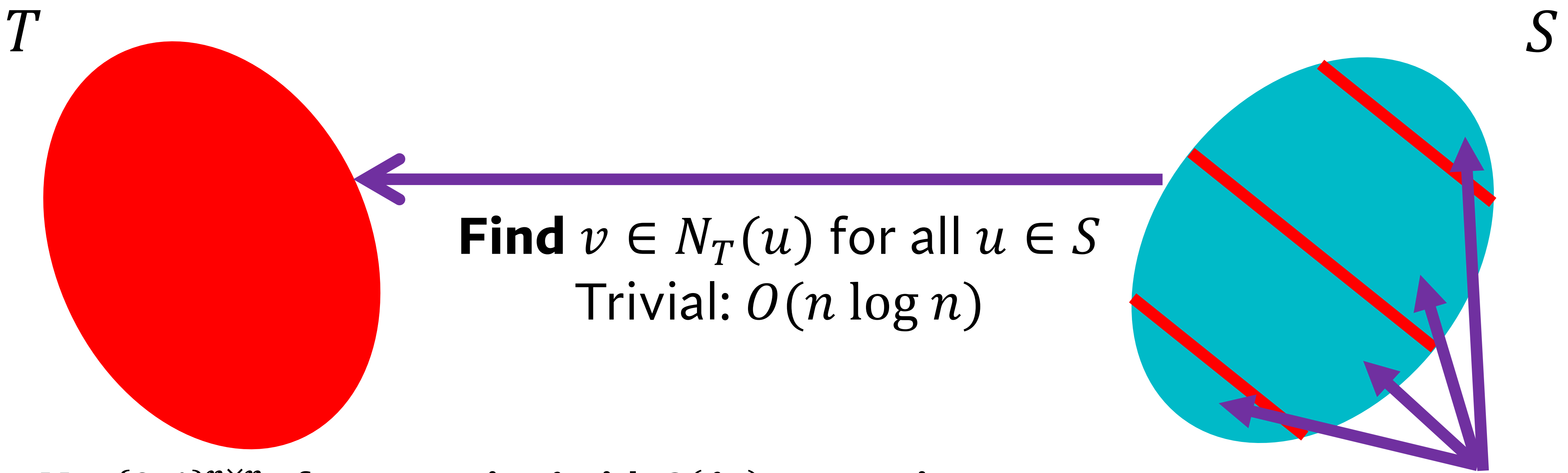


Seperating Matrices: Learn $M \in \{0,1\}^{n \times n}$ of row sparsity ℓ with $O(\ell n)$ cut queries

- **Trivial:** $O(\ell n \log n)$ cut queries using binary search.
- A cut query provides $\Omega(\log n)$ bit of information.
- Can be used to shave the log factor.

The (Easy) Problem

Each $u \in S$ **finds** a neighbor $v \in T$



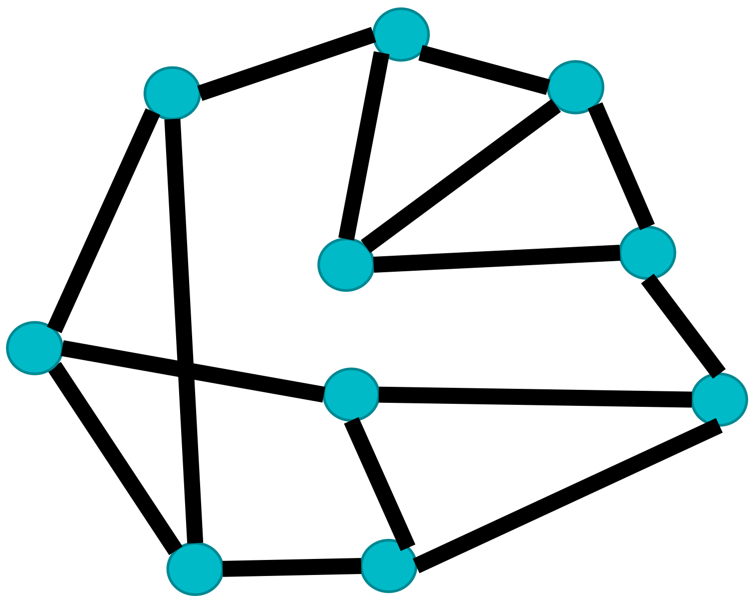
- **Learn $M \in \{0,1\}^{n \times n}$ of row sparsity ℓ with $O(\ell n)$ cut queries**
- **Sampling Lemma**: Solve **The (Easy) Problem** in $O(|S|)$ cut queries, **in expectation**
- Immediate corollary: **Connectivity** in **expected $O(n)$** cut queries!
- Lemma solves **The Problem** as well in **expected $O(n)$** cut queries!

Degree buckets $[2^i, 2^{i+1}]$
Sample w.p. $\frac{1}{2^i}$ for $O(1)$ sparsity

Remains (α, β)-good

Beyond $n \log n$ step 1: Refined star contraction

- $|R|$ and therefore $|V'|$ are too large
- Replace $p = \Theta\left(\frac{\log n}{\delta}\right)$ with $p = \Theta\left(\frac{\log \delta}{\delta}\right)$
- **Immediate**: $|R| = O\left(\frac{n \log \delta}{\delta}\right)$
- $|V^*| = |\{v \in V \mid N(v) \cap R = \emptyset| = O\left(\frac{n}{\delta}\right)$
- $V' = R \cup V^*$
- Each $u \notin V'$, independently samples neighbor $v \in N_R(u)$
- Contract into $G' = (V', E')$
- $O(n \log \delta) \Rightarrow O(n \log \log n)$ queries.



$O(n)$ expected queries

Solved!

If $\delta \geq polylog(n)$, use **[Mukhopadhyay, Nanongkai 2020]**

Beyond $n \log \log n$

- Need to have $O\left(\frac{n}{\delta}\right)$ vertices within $O(n)$ complexity.
- From $O\left(\frac{n \log \delta}{\delta}\right)$ to $O\left(\frac{n}{\delta}\right)$:
    - Learn a dense enough subgraph with $O(n)$ queries.
    - Do 2-out contraction within it.

Summary

Theorem. Randomised cut-query algorithm for min-cut in simple graphs has $O(n)$ complexity.

Open Questions:

- Randomised communication complexity of edge connectivity?
  - SOTA: $\Omega(n \log\log n)$ [AD21]
- Zero-error/Deterministic edge connectivity with $O(n)$ cut queries?
- Weighted graph: $O(n)$ cut query?
- In general: SFM needs $\omega(n)$ evaluation query accesses?

Thank you!