Deterministic Edge Connectivity and Max Flow using Subquadratic Cut Queries

Aditya Anand

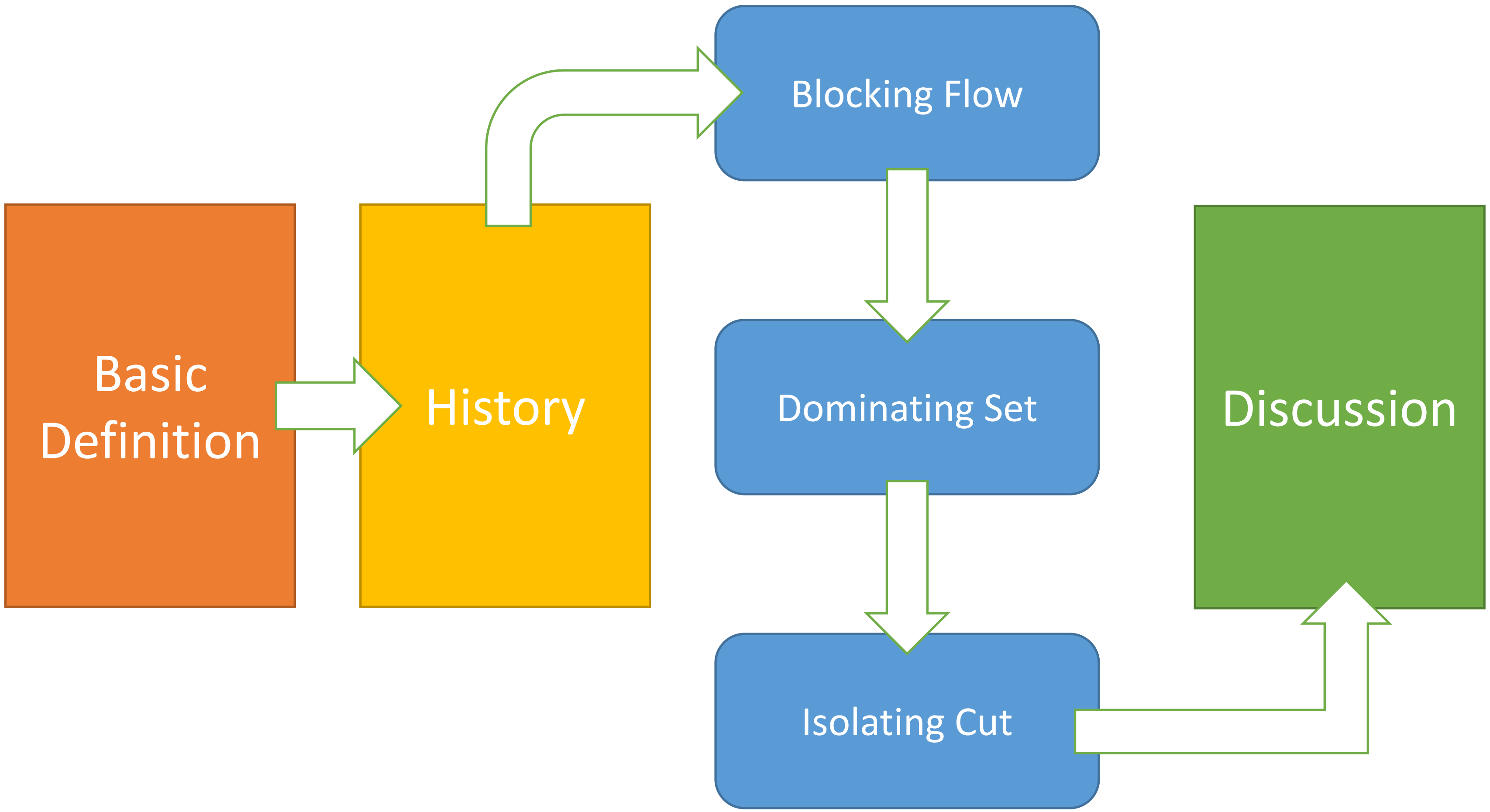
Thatchaphol Saranurak

University of Michigan

Yunfan Wang

Tsinghua University, IIIS

Outline



Basic Definition
History
Blocking Flow
Dominating Set
Isolating Cut
Discussion

Basic Definition

- **Cut Query Model**

Q: How many **query** do we need to compute the **global minimum cut**?

Sense of the problem

"Learn the whole graph"

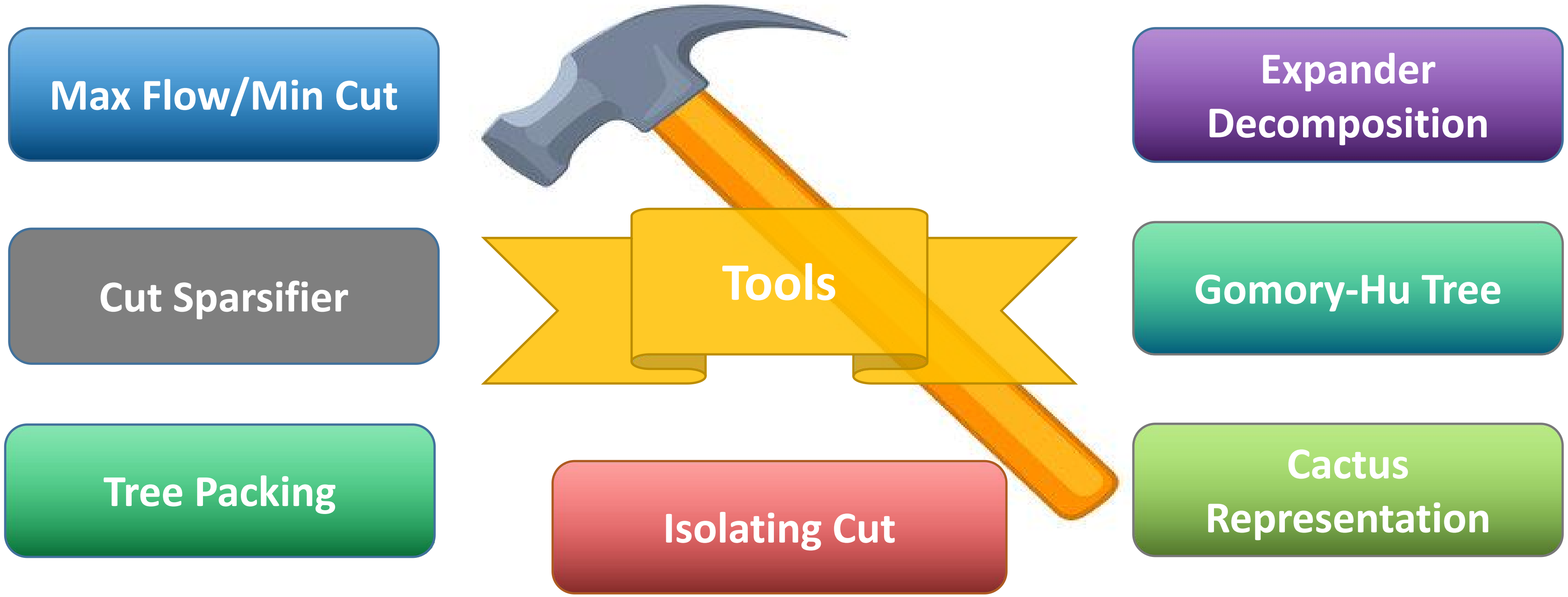
- **A Naive Algorithm**
- Learn each edge (u, v) by computing $\mathrm{Cut}(\{u\}) + \mathrm{Cut}(\{v\}) - \mathrm{Cut}(\{u, v\})$, which requires $O(n^2)$ queries [**Nearly Optimal**]

- For dense graph, one needs at least $\sim\Omega(n^2)$ **query** to learn the whole graph [Rubinstein, Schramm & Weinberg, ITCS 2018]

- One can use $O(n^2/\log n)$ **query** to learn the whole graph[Grebinski & Kucherov, Algorithmica 2000]

Motivation

- **Q**: Can we use **less information** to **represent** the **global minimum cut**?



History

- **Edge Connectivity** in **Cut Query Model**

	Det./ Rand.	Simple/ Weighted	Query
Rubinstein, Schramm& Weinberg[RSW18]	Rand.	Simple	$O(n \log^3 n)$
Mukhopadhyay & Nanongkai[MN20]	Rand.	Weighted	$O(n \log^{O(1)} n)$
Apers et al.[AEG+22]	Rand.	Simple	$O(n)$
This paper	Det.	Simple	$\tilde{O}(n^{\frac{5}{3}})$

State-of-the-Art

- Different Settings in **Cut Query Model**

	Connectivity		Edge Connectivity	
	Lower	Upper	Lower	Upper
Deterministic	$\Omega(n)$ [HMT88]	$O(\frac{n \log n}{\log \log n})$ [LC24]	$\Omega(n)$ [HMT88]	$\tilde{O}(n^{5/3})$ **(This paper)**
Zero-error, Randomized	$\Omega(\frac{n \log \log(n)}{\log n})$ [RS95]	$O(n)$ [AEG+22]	$\Omega(n)$	$\tilde{O}(n^{5/3})$ **(This paper)**
Bounded Error, Randomized	$\Omega(\frac{n}{\log n})$ [BFS86]	$O(n)$ [AEG+22]	$\Omega(\frac{n \log \log(n)}{\log n})$ [AD21]	$O(n)$ [AEG+22]
Quantum	$\Omega(1)$	$O(\log^5(n))$ [AL21]	$\Omega(1)$	$\tilde{O}(\sqrt{n})$ [AEG+22]

Max Flow/Min Cut

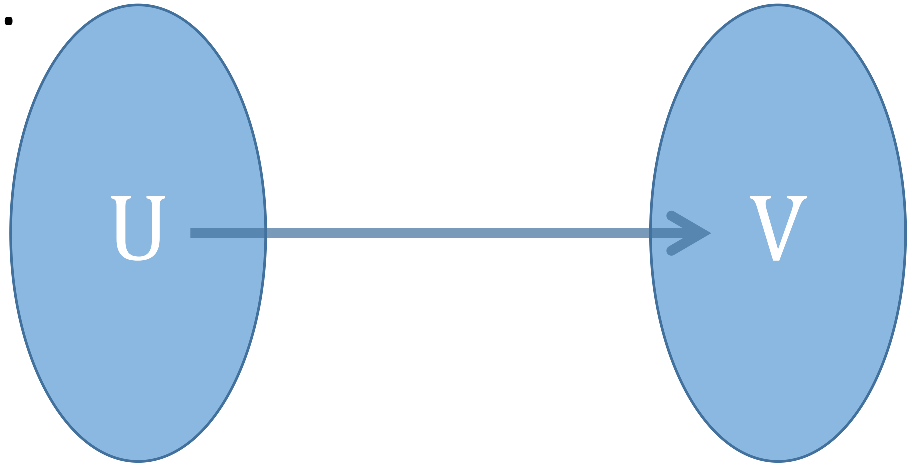
- **Max Flow Min Cut Theorem**
- No Duality Gap for $s-t$ max flow and $s-t$ min cut

max flow value=min cut capacity

- Q: Why should we consider **Max Flow** as a start point?
- Inspiration: [RSW18] shows that, for graph G with integral weights from $[0, W]$, every $s-t$ flow of value f can be covered by edges of at most $O(n\sqrt{fW})$ total weight
- **We can use $O(n\sqrt{n})$ edges to cover any s-t flow in simple graph!**

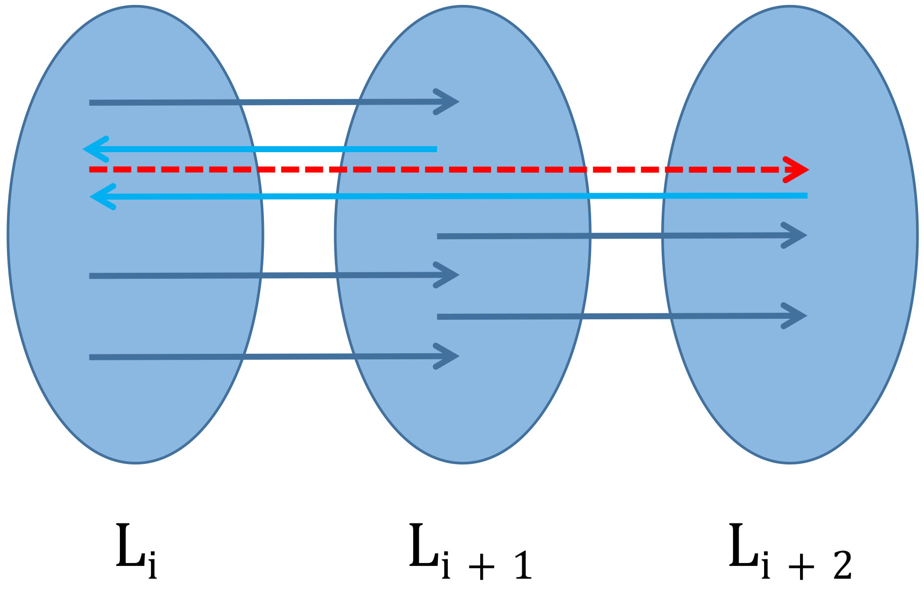
BIS/Cross Query

- A **BIS (Bipartite Independent Set)** or **Cross Query** asks whether there exists an edge between two sets U and V. In other words, it checks if there is an edge (u, v) such that $u \in U, v \in V$.
- **Fact:** A **BIS/Cross query** can be replaced by $O(1)$ Cut Query in undirected graph.

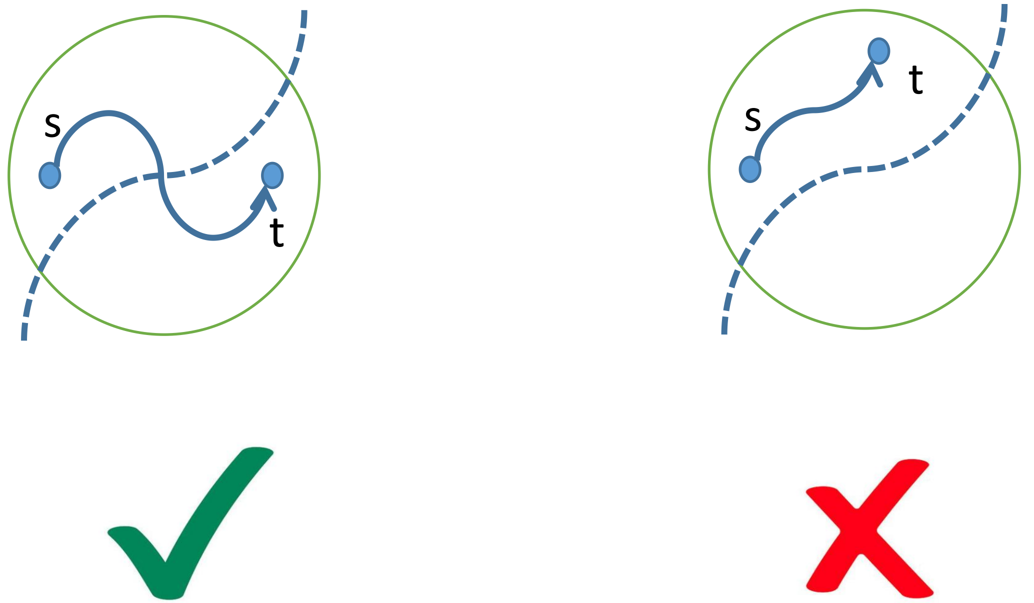


Blocking Flow

- **Theorem 1**: We can use $\sim O(n^{5/3})$ **BIS/Cross query** to obtain an explicit s-t max flow in simple graph [**Idea: Dinitz'algorithm**]

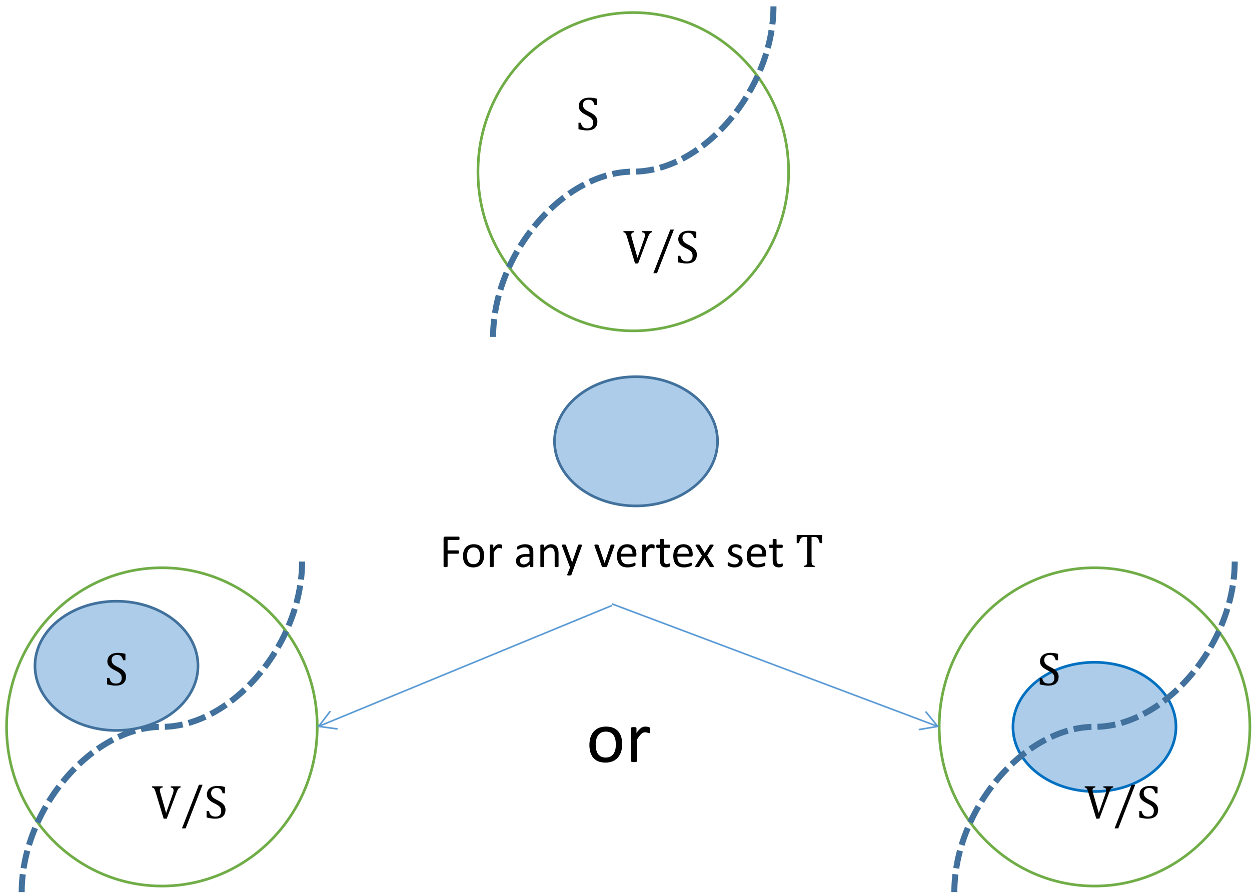


From Flow to Cut



How can we guarantee that s and t are on **different** sides of the minimum cut?

“Preserve the Minimum Cut”

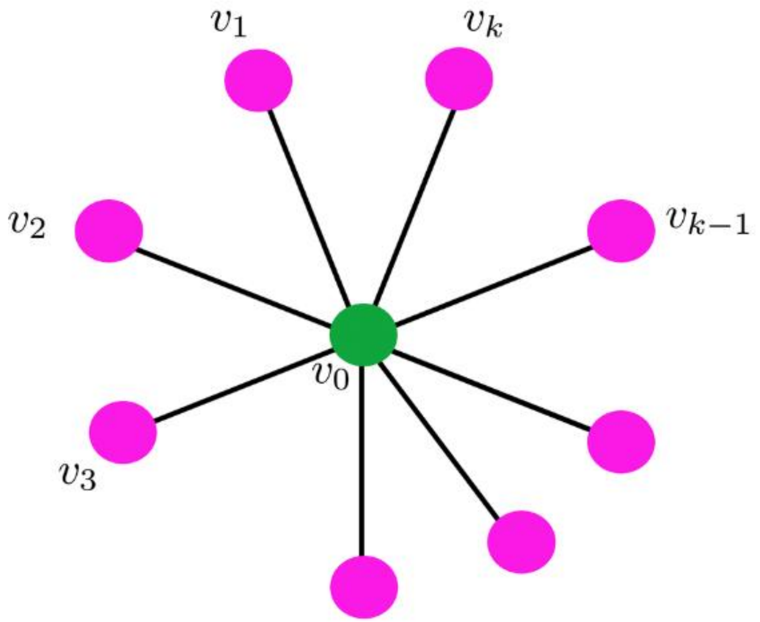


Contraction
[e.g. Karger’s Algorithm]

“Preserve the Min Cut”
[e.g. Kawarabayashi & Thorup, STOC 2015]

Dominating Set

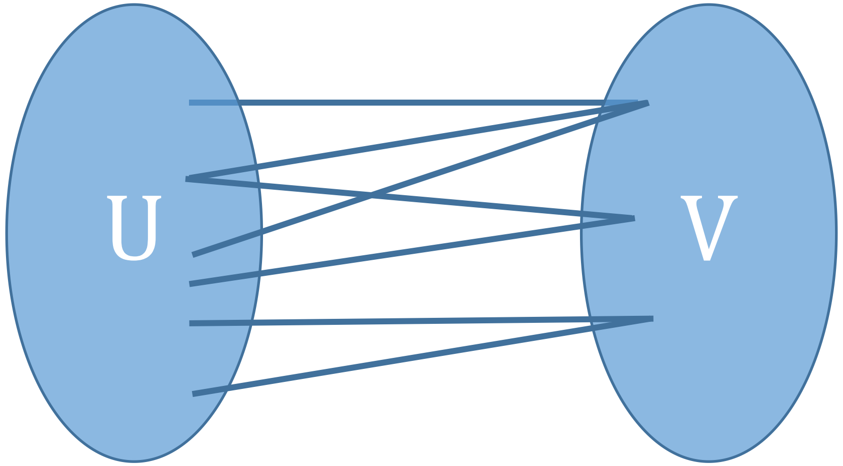
- **A Key Observation:** In simple graph, a dominating set can "preserve all non-trivial minimum cut"



Special Case: Star Graph[No non-trivial minium cut]

Dominating Set

- **Theorem 2**: If the minium degree is δ, then we can find a dominating set D with size at most $O(\frac{n}{\delta}\log\frac{n}{\delta})$ with $\sim O(n)$ cut **query**.
- Existence: Sample each vertex with probability $\sim\frac{1}{\delta}$.
- **De-randomize Idea:** Finding an element **above the average**

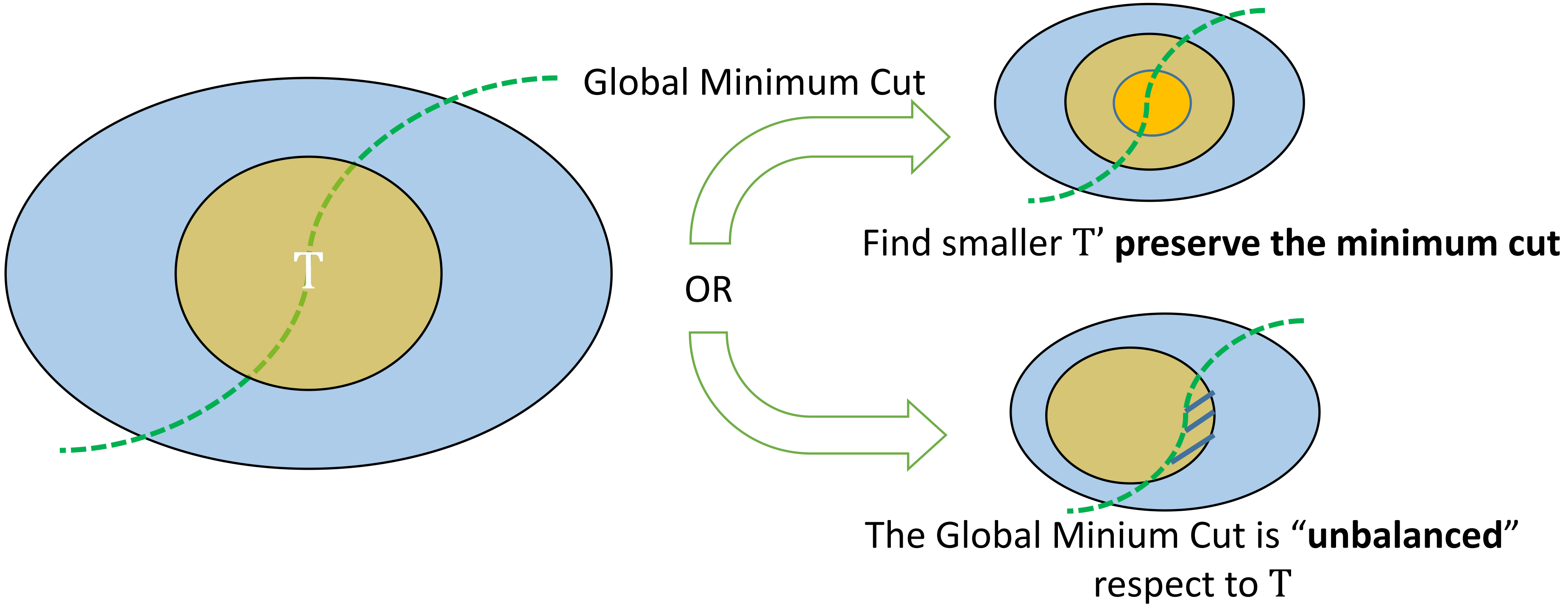


In $O(\log n)$ queries

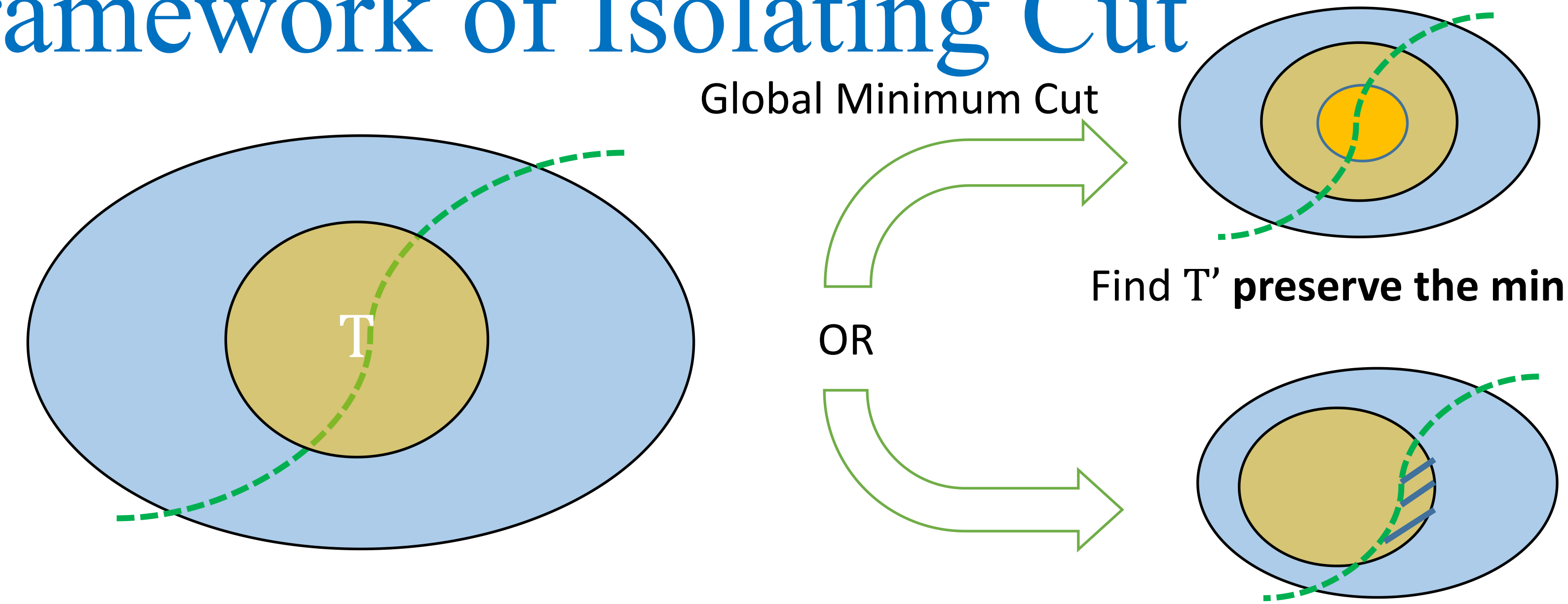
Find $u \in U$ such that $\deg_V(u) \geq |E(U,V)|/|U|$

Framework of Isolating Cut

- Isolating Cut[Li & Panigrahi, FOCS 2020]



Framework of Isolating Cut



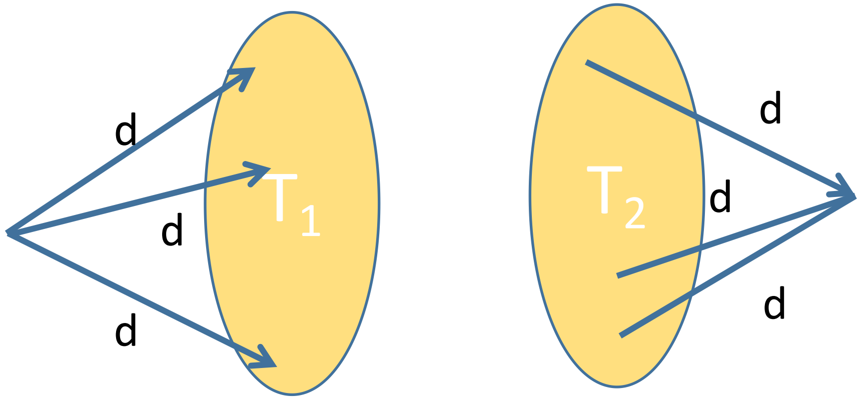
Global Minimum Cut

Find T' **preserve the minimum cut**

OR

The Global Minium Cut is "**unbalanced**"

Cut Matching Game



Almost every edge saturated

OR

Find balanced sparse cut

Polylog times

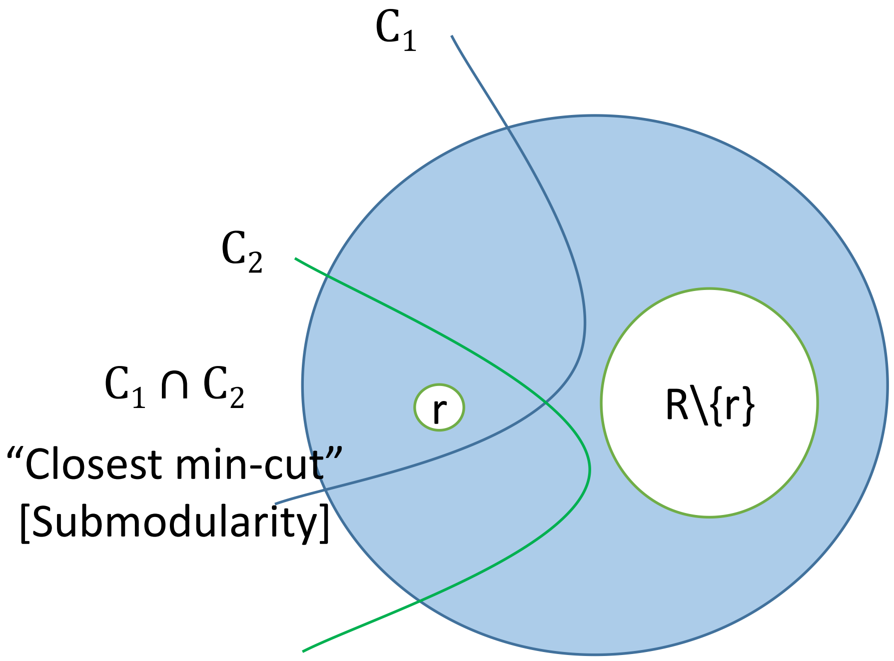
Guarantee "Almost Expander"

Demand $d \leq \delta$ **[Total Flow Size** $\leq \sim O(n)$ **]**

NOT exactly the same as [LP20], we don't compute $T_1 - T_2$ min-cut

Minimum Isolating Cut

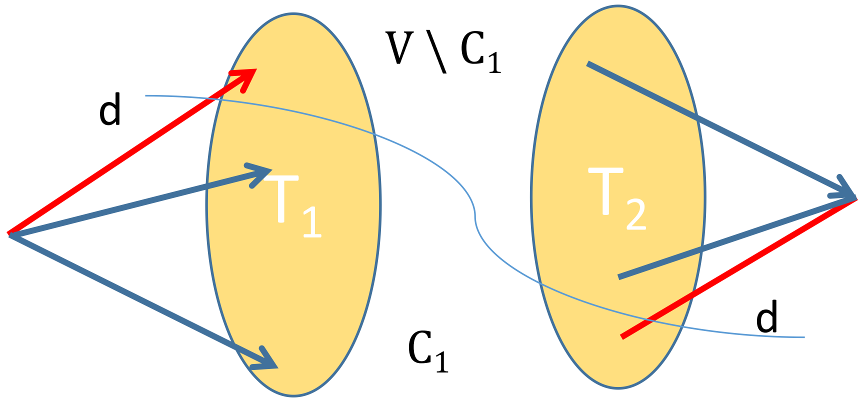
- For any set of vertices $R \subseteq V$, $r \in R$, the minimum isolating cut of r is an $\{r\} - \{R \setminus \{r\}\}$ min-cut



Subroutine

- Let $d = \tau + 1$, we will either outputs an **isolating cut** of R of size at most τ,
- or certifies that the **minimum isolating cut** of R has a size larger than τ.

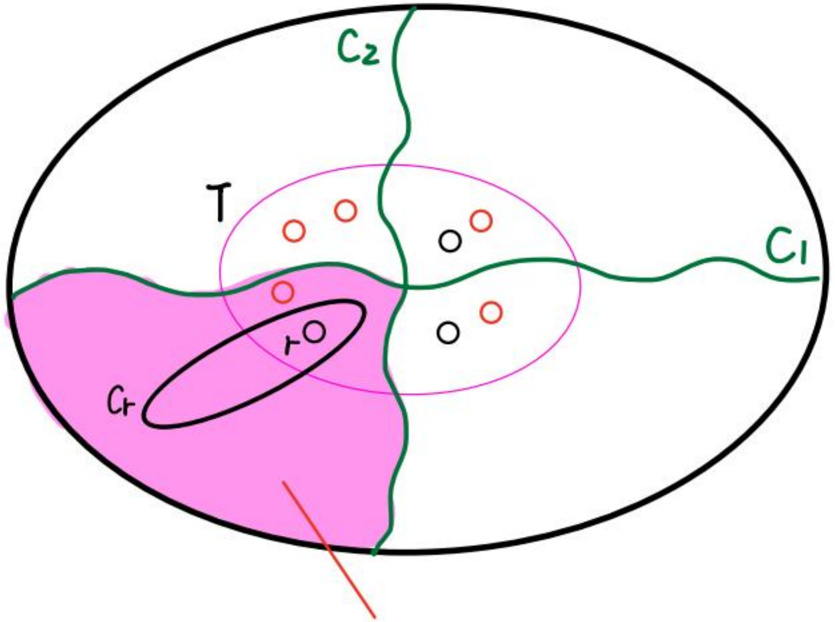
Cut Matching Game



If an edge is **saturated**, then the corresponding minimum isolating cut has size at least d.

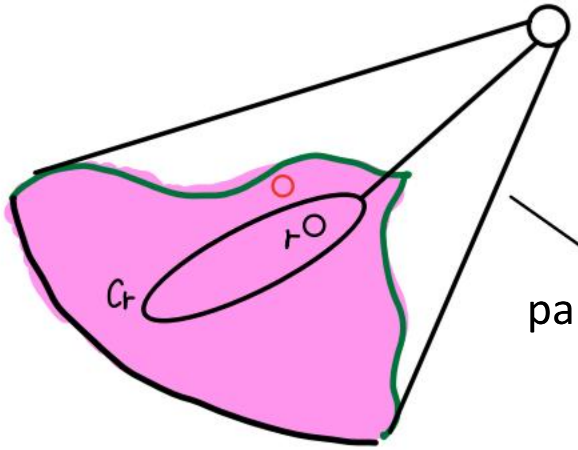
If the minimum isolating cut of R is C_r less than d, then we must have $C_r \subseteq C_1$ or $C_r \subseteq V/C_1$

“From Global to Local”



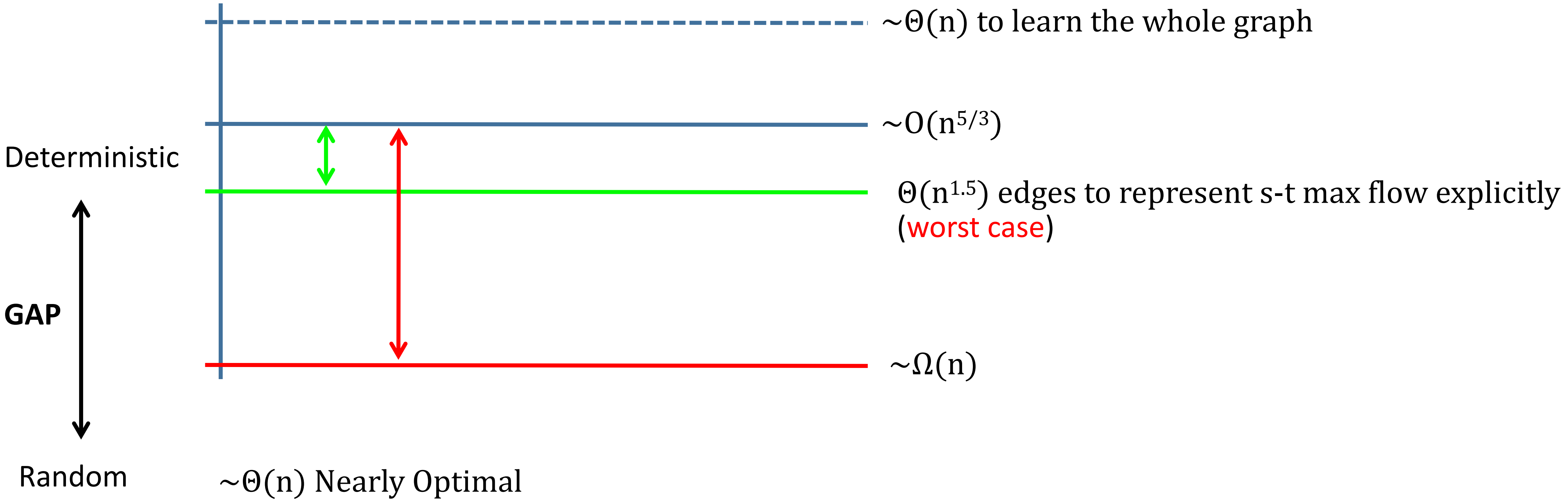
- not saturated in all max-flow call
- saturated at least once

Guarantee: each part contains at most 1 black vertex



Call $r - S(r)$ max flow

Discussion


~Θ(n) to learn the whole graph
~O(n^{5/3})
Deterministic
Θ(n^{1.5}) edges to represent s-t max flow explicitly
(worst case)
GAP
~Ω(n)
Random
~Θ(n) Nearly Optimal