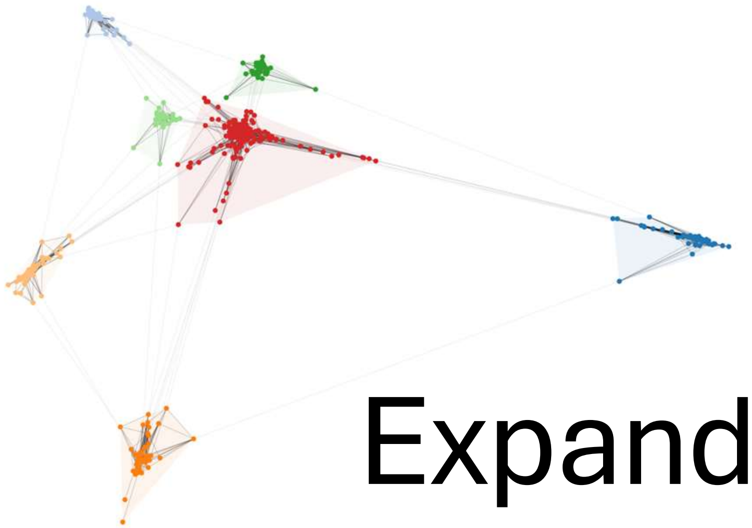


Lecture 1: Expander Decomposition

Thatchaphol Saranurak
U of Michigan

August 18, 2025
ADFOCS

Part 0
Setting expectation

Expanders in TCS:

There are 2 main different regimes of expanders in TCS

1. **Tailor-made expanders**
    - Strong expansion
    - Non-trivial to even construct explicitly.
2. **Expanders in the wild**
    - Weaker expansion
    - Can find everywhere

Tailor-made expanders

- **Goal**: Explicit construction of extremely strong expanders
- **Key objects**:
  - Ramanujan expanders
  - Lossless expanders, monotone expanders, more
  - High dimensional expanders
- **Main applications**:
  - Coding theory
  - Pseudo-randomness (extractors, condensers, dispersers)
  - PCP construction
  - Sampling algorithms

Expanders in the wild

- **Goal:** Find and use expanding subsets in an arbitrary graph
- **Key objects**:
  - Expander Decomposition
  - Expander Hierarchies
- **Main applications:**
  - Graph theory (grid minor theorem, edge disjoint paths)
  - Graph algorithms (max flow, mincut, sparsifiers, oblivious routing)
  - Dynamic / Fault-tolerant data structures (connectivity, distance)

This series is about expanders in this regime

Topics for 5 lectures

Lecture 1	**Expander decomposition**
Lecture 2,3	**Two types of expanding hierarchies**
Lecture 4,5	**Overview of whole area**

Expectation

You will learn:

- Intuition of the structure of expander decomposition/hierarchy
  - Unified view $\Rightarrow$ you can navigate the literature much easier
- Algorithms and data structures based on them

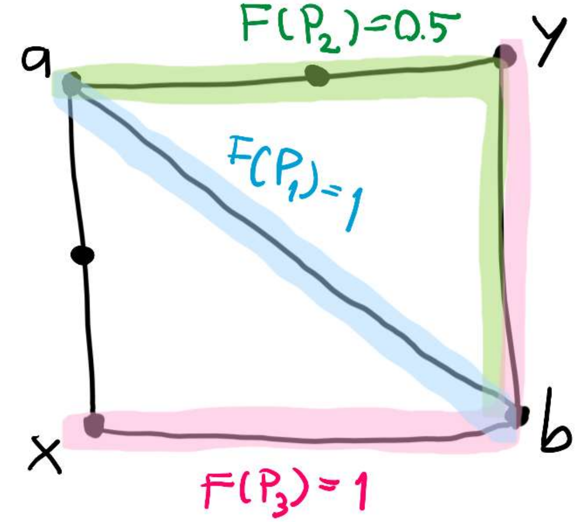
Omit:

- Fast algorithms for computing expander decomposition/hierarchy
- See my videos on Expanders and Fast Graph Algorithms

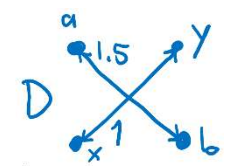
Part 1
Basic Definitions

Flow and Demands

- In this talk, graph $G = (V, E)$ is always **undirected**
- (Multi-commodity) flow F
  - assigns flow value $F(P)$ on path P
  - Congestion:
    - $\text{cong}_F(e) = F(e)/\text{cap}(e)$
    - $\text{cong}(F) = \max_e \text{cong}_F(e)$
- Flow F routes **demand** D if
  - $D(a, b) = \Sigma_{(a,b)-\text{pa}\quad P} F(P)$ for all (a, b)
  - Think of D as a capacitated graph
- Demand D is **routable with congestion κ** if
  - $\exists F$ routing D with $\text{cong}(F) = \kappa$
  - Say "D is **routable**" if $\kappa \leq 1$

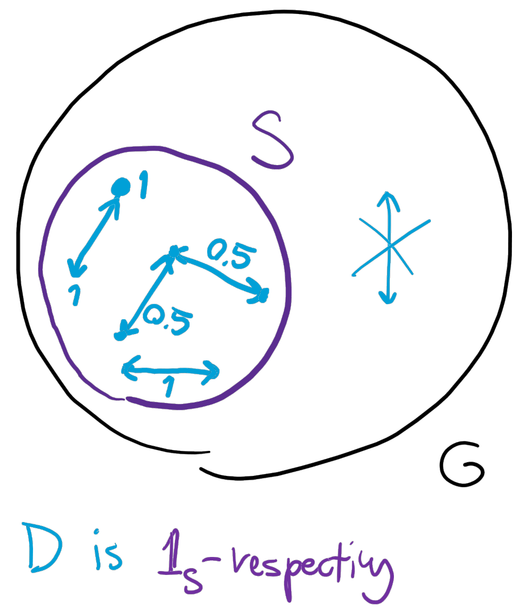


Example:
- $\text{cong}(F) = 1.5$
- F routes D such that $D(a, b) = 1.5$, $D(x, y) = 1$



Node-Weighting

- Demand D is ***A*-respecting** if
  - $\deg_D(v) \coloneqq \Sigma_{(v,w)} D(v) \leq A(v)$ for all v
- We call A a **node-weighting**
  - $|A| \coloneqq \Sigma_v A(v)$
  - $A(S) \coloneqq \Sigma_{v \in S} A(v)$
  - $A \cap S$ is such that $(A \cap S)(v) = \begin{cases} A(v) & \text{if } v \in S \\ 0 & \text{if } v \notin S \end{cases}$
- Key examples:
  - $A = 1_S$ for $S \subseteq V$
  - $A = \deg_G$
  - $A = \deg_F$ for $F \subseteq E$



Part 2
Expansion

Two equivalent ways to think about expansion

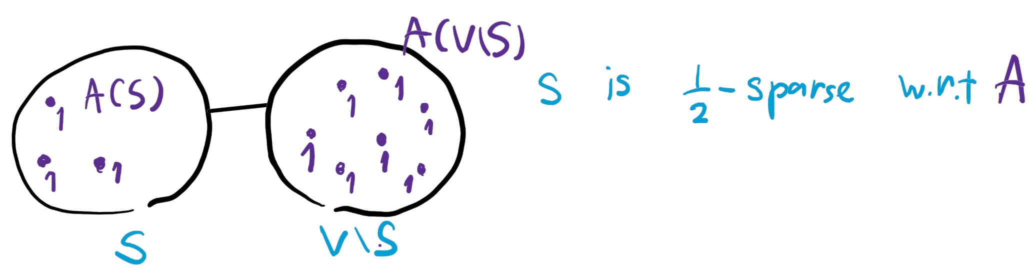
Flow and Cut Expansions: **Informal**

- A is **flow-expanding in G** if
  - can route flow between A with low congestion
- A is **cut-expanding in G** if
  - No bottleneck cut preventing routing flow between A with low congestion

Flow and Cut Expansions: **Formal**

- A is $\boldsymbol{\phi}$**-flow-expanding in G** if
  - Every A-respecting demand is routable in G with congestion $1/\phi$
  - $\Leftrightarrow$ Every $(\phi \cdot A)$-respecting demand is routable in G

- A is $\boldsymbol{\phi}$**-cut-expanding** in G if
  - For every set $S \subset V$, $\mathrm{cap}(S, V \backslash S) \geq \phi \min\{A(S), A(V \backslash S)\}$
  - S is a ϕ-sparse cut w.r.t. A if $\mathrm{cap}(S, V \backslash S) < \phi \min\{A(S), A(V \backslash S)\}$
  - A is not $\boldsymbol{\phi}$**-cut-expanding** $\Leftrightarrow$ no ϕ-sparse cut w.r.t. A

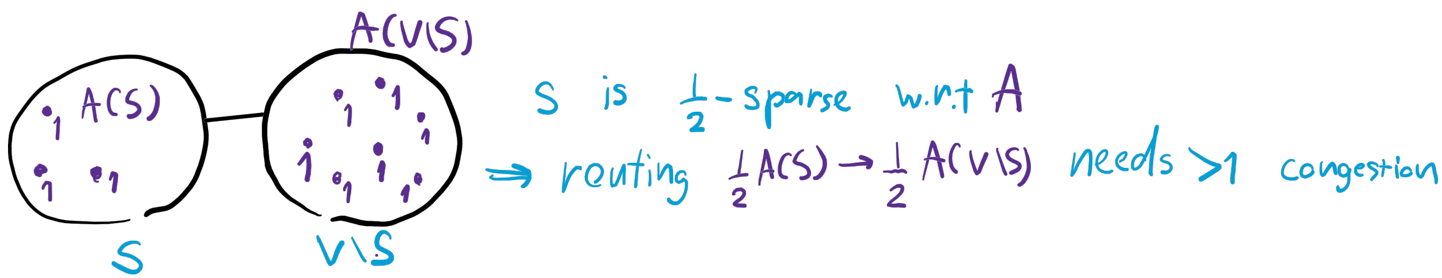


Flow and Cut Expansions: **Equivalence**

Fact:

if A is ϕ-flow-expanding in $G \Rightarrow A$ is ϕ-cut-expanding in G

- **Proof**: suppose not. $\exists S$ where $\mathrm{cap}(S, V \backslash S) < \phi \min\{A(S), A(V \backslash S)\}$.
- Then, $\exists (\phi A)$-respecting demand require congestion > 1.



[Leighton Rao'88]:

if A is ϕ-cut-expanding in $G \Rightarrow A$ is $\frac{\phi}{\log n}$-flow-expanding in G

“Expanding”

- Think: **flow-expanding** $\approx$ **cut-expanding**
- Will say “expanding” for both
  - Ignore the $\log n$ factor loss

- When we say “expanding” without ϕ, think of $\phi \geq 1/\text{polylog}(n)$

Expanders and Expanding Edge Sets

Def: G is a ϕ-expander $\Leftrightarrow$ $\deg_G$ is ϕ-expanding in G

- **Intuition:** "reasonable" demand is routable with congestion $1/\phi$
- "Reasonable" demand = $\deg_G$-respecting demand.
  - To route with congestion 1, we must respect the vertex degree.

Def: $F \subseteq E$ is ϕ-expanding in G $\Leftrightarrow$ $\deg_F$ is ϕ-expanding in G

When G has many connected components

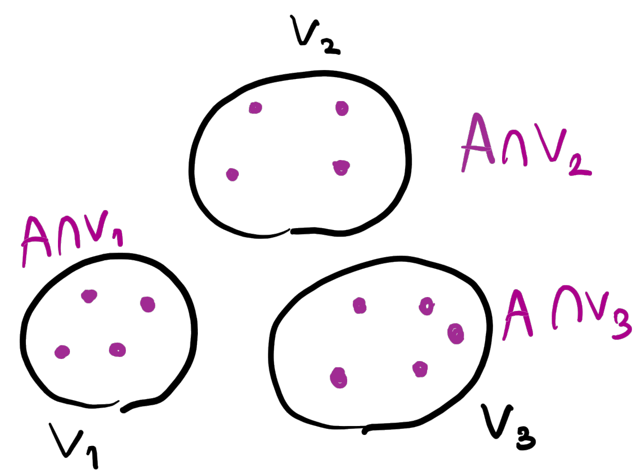
Suppose G has many connected components.

Def: A is ϕ-expanding in $G \iff$

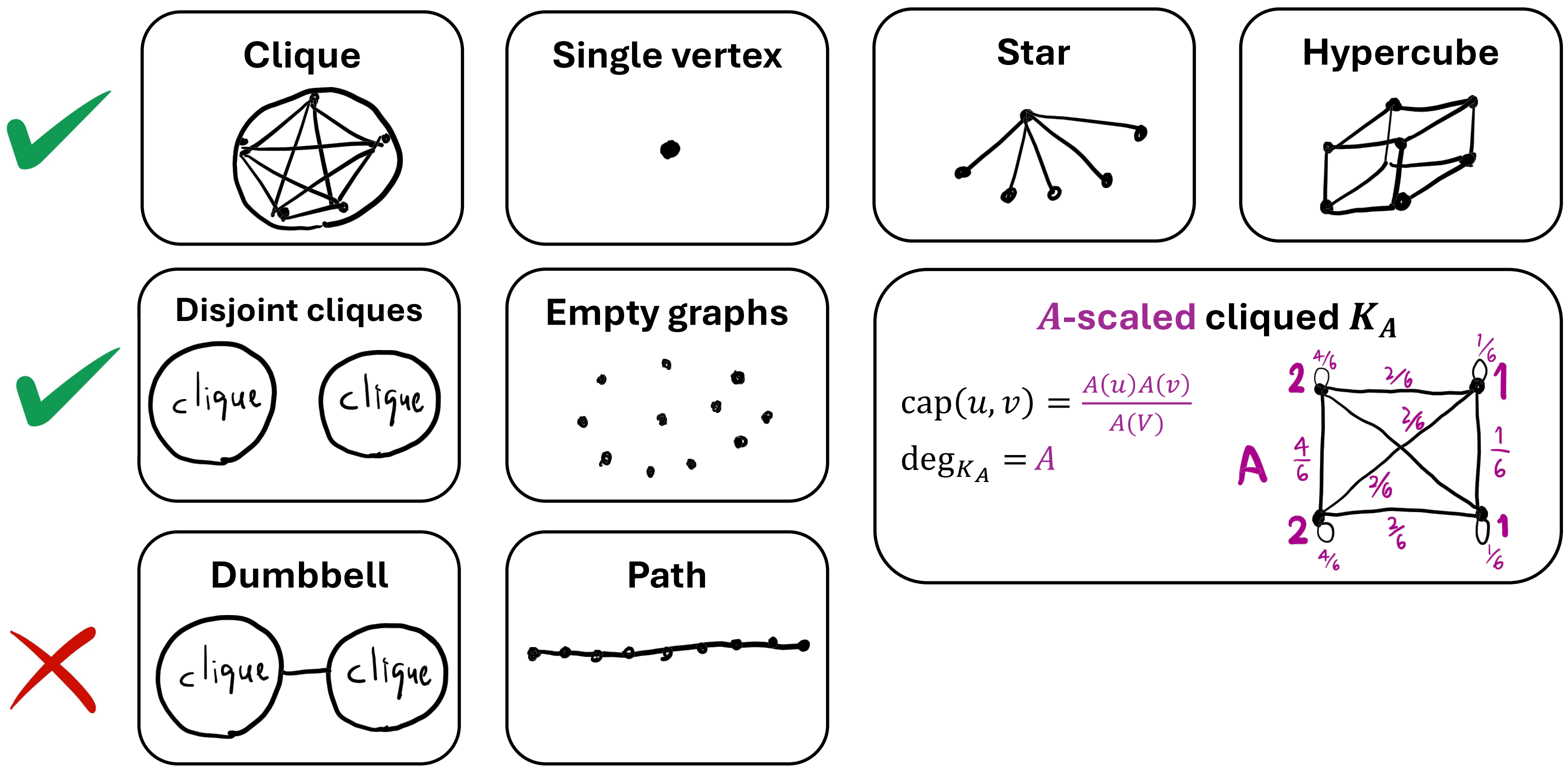
for each component U in G, $A \cap U$ is ϕ-expanding in G

Def: G is ϕ-expander $\iff$

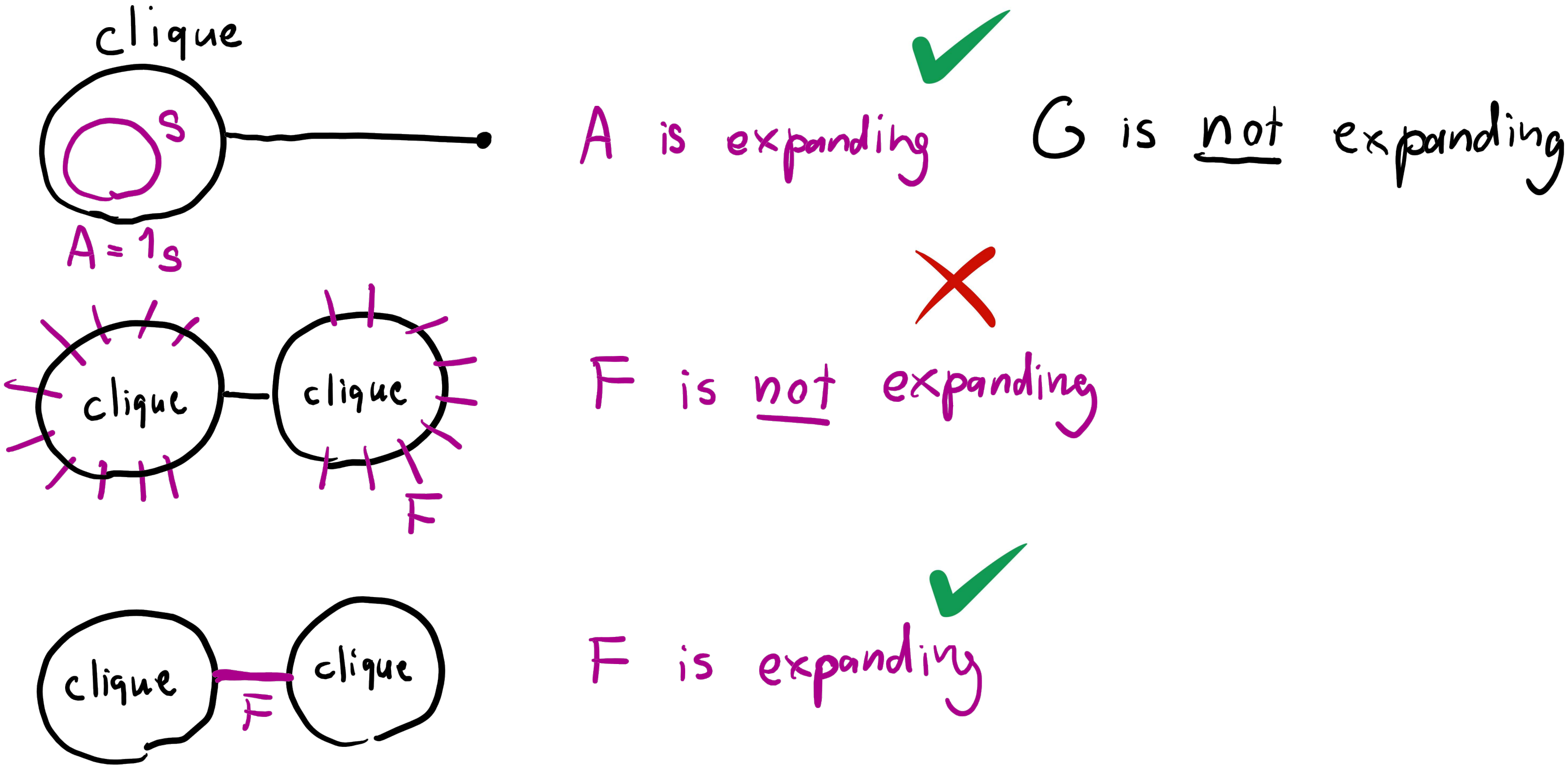
every component of G is ϕ-expander



Quiz: which one is an expander?



Quiz: which set is expanding?



Quiz

Suppose A is ϕ-expanding in G.

Are these true?

- A is ϕ-expanding in $G' \supseteq G$.
- For any $A' \leq A$, A' is ϕ-expanding in G.
- $2A$ is $\phi/2$-expanding in G.

Part 3

Algorithms on Expanders

Expanders are Algorithmic Friendly

Problems usually become easy on expanders

You will see many examples in this series.

Example: Approx Max Flow on Expanders

On ϕ-expander, can ϕ-approximate (s,t)-maxflow $\lambda_{s,t}$ in $O(1)$ time.

$$\phi \min\{\deg(s), \deg(t)\} \leq \lambda_{s,t} \leq \min\{\deg(s), \deg(t)\}$$

- $\lambda_{s,t} \leq \min\{\deg(s), \deg(t)\}$ as $\{s\}$ and $\{t\}$ are (s,t)-cuts
- $\lambda_{s,t} \geq \phi \min\{\deg(s), \deg(t)\}$
  - Demand D where $D(s,t) = \min\{\deg(s), \deg(t)\}$
  - D respects $\deg_G \Rightarrow D$ is routable with congestion $1/\phi$.
  - $\Rightarrow \exists\ (s \to t)$ flow of size $\phi D(s,t)$ with congestion 1

Part 4
Expander Decomposition

Motivation

G might not be an expander, but…

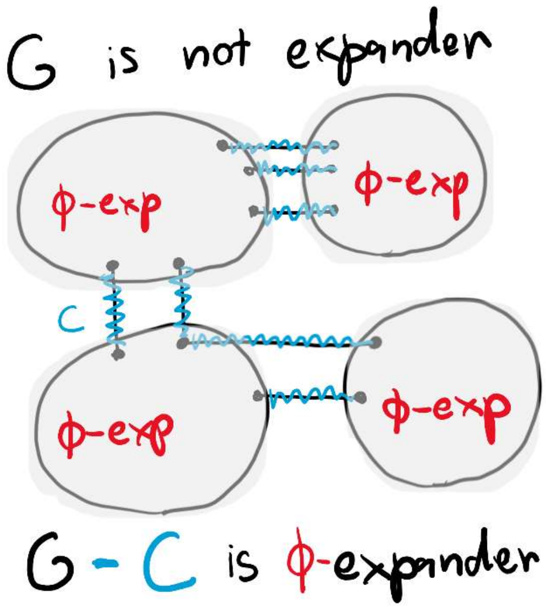
We can make G a ϕ-expander after removing $\approx \phi$ fraction of edges

ϕ-expander decomposition of G

Theorem: Given $G = (V, E)$, ϕ, there exists $C \subseteq E$

- $|C| \le (\phi \log n) \cdot m$
- $\deg_G$ is ϕ-expanding in $G - C$.

So, $G - C$ is a ϕ-expander



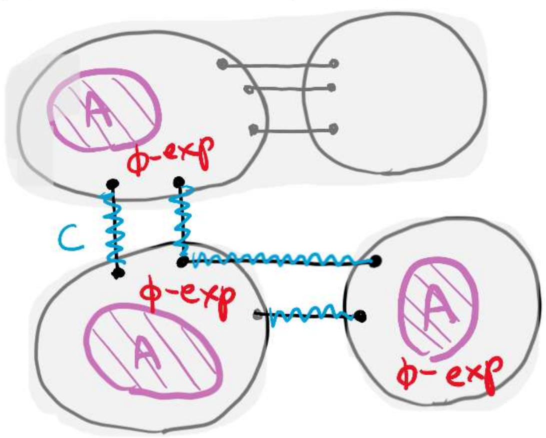
C "decompose" graph G so that for each component U in $G - C$, $G[U]$ is an expander

ϕ-expander decomposition of A in G

Theorem: Given $G = (V, E)$, A, ϕ, there exists $C \subseteq E$

- $|C| \leq (\phi \log n) \cdot |A|$
- A is ϕ-expanding in $G - C$

A is not expanding in G



A is ϕ-expanding in G-C

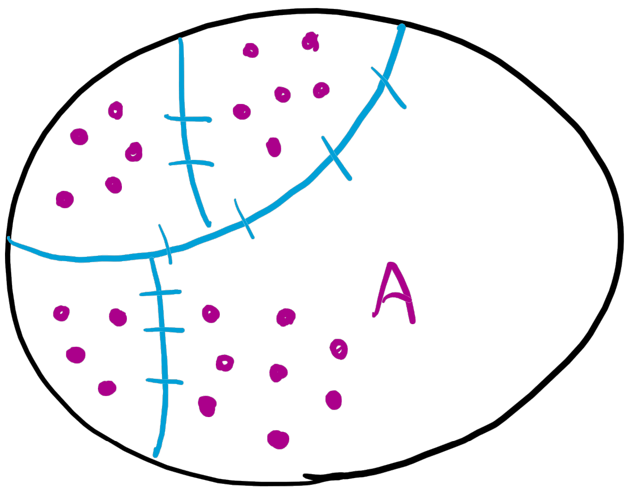
C "decompose" graph G so that for each component U in $G - C$, $A \cap U$ is expanding in $G[U]$

Algorithm

- $C \leftarrow \emptyset$
- While A is not ϕ-expanding in $G - C$
  - So, $\exists$ ϕ-sparse cut $(S, U - S)$ in component U of $G - C$
    $$|E(S, U - S)| < \phi \min\{A(S), A(U - S)\}$$
  - $C \leftarrow C \cup E(S, U - S)$
- Return C

Analysis: After terminated

- A is ϕ-expanding in $G - C$ ($A \cap U$ is expanding in $G[U]$ $\forall U$)
- Remain to bound $|C|$



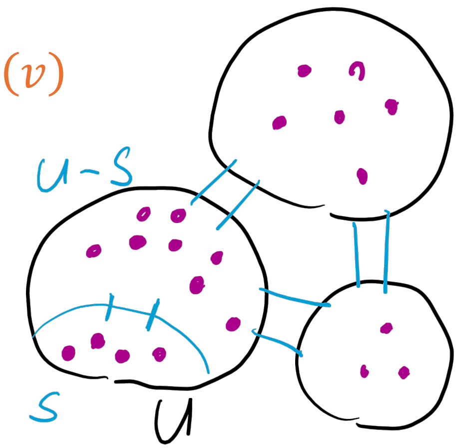
Bound $|C|$

Plan:

- Initially, each A-vertex has $\$(\phi A(v) \log n)$
- Pay \$1 per edge in C without debt
- $\Rightarrow |C| \leq \phi|A| \log n$

When $C \leftarrow C \cup E(S, U - S)$, each A-vertex in the smaller side pays $\$\phi A(v)$

- Total Budget: $\$\phi \min\{A(S), A(U - S)\}$
- Total Cost: $\$E(S, U - S)$
- Cost $\leq$ Budget as S is ϕ-sparse $(E(S, U - S) \leq \phi \min\{A(S), A(U - S)\})$



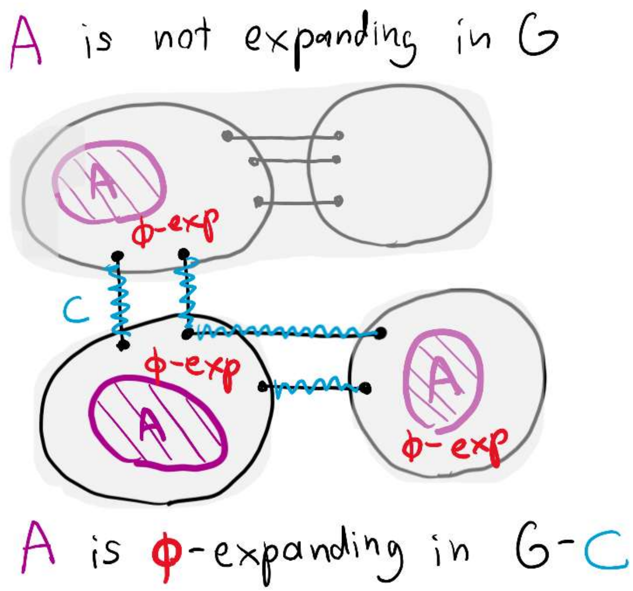
Each vertex has $\geq \$0$ at all time

- A vertex is put to the smaller side $\leq \log n$ times

ϕ-expander decomposition of A in G

Theorem: Given $G = (V, E), A, \phi$, there exists $C \subseteq E$

- $|C| \leq (\phi \log n) \cdot |A|$
- A is ϕ-expanding in $G - C$

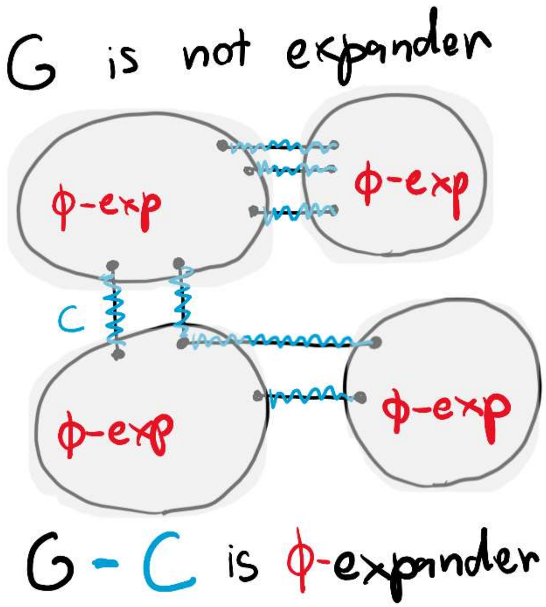


Will call C an **ϕ-ED** of A in G

ϕ-expander decomposition of G

Theorem: Given $G = (V, E)$, ϕ, there exists $C \subseteq E$

- $|C| \leq (\phi \log n) \cdot m$
- $\deg_G$ is ϕ-expanding in $G - C$.



Will call C an **ϕ-ED** of G

Part 5

Repeated Expander Decomposition

Repeated Expander Decomposition

Idea:

Compute an expander decomposition $\mathcal{C}$ of G.

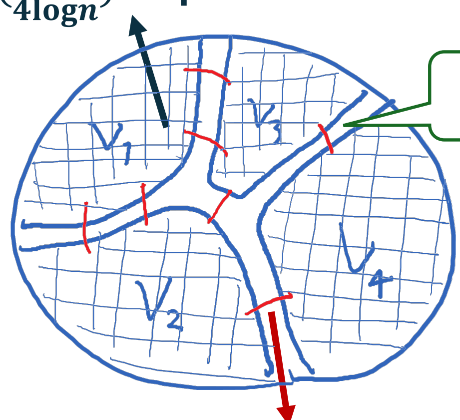
Then, recurse on the graph induced by $\mathcal{C}$.

Repeated Expander Decomposition

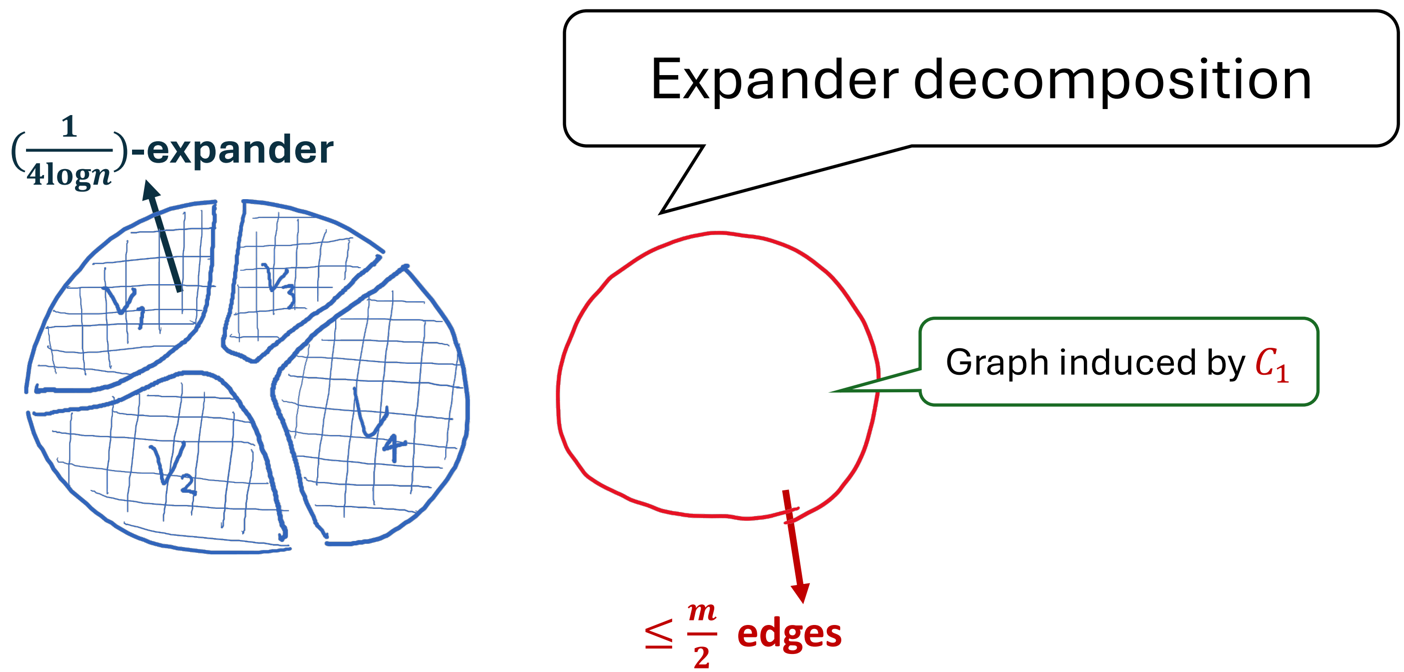
Theorem: Given $G = (V, E)$, can partition E

- Each part induces a $(\frac{1}{4\log n})$-expander
- Each vertex is in $\leq \log n$ expanders

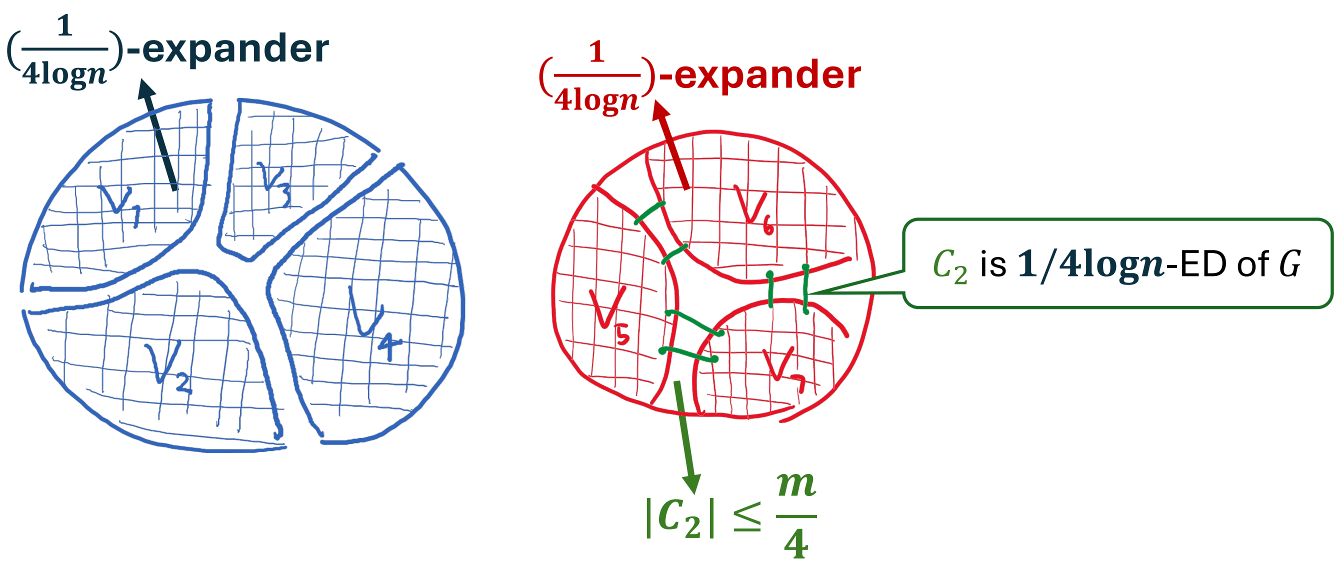
Repeated Expander Decomposition



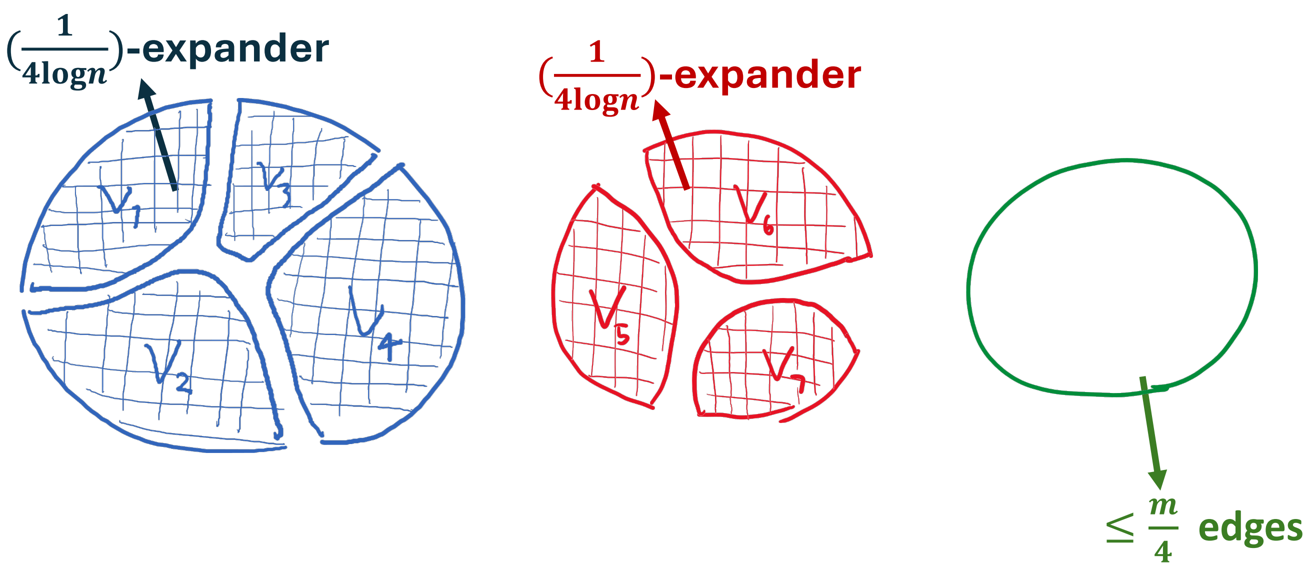
Repeated Expander Decomposition



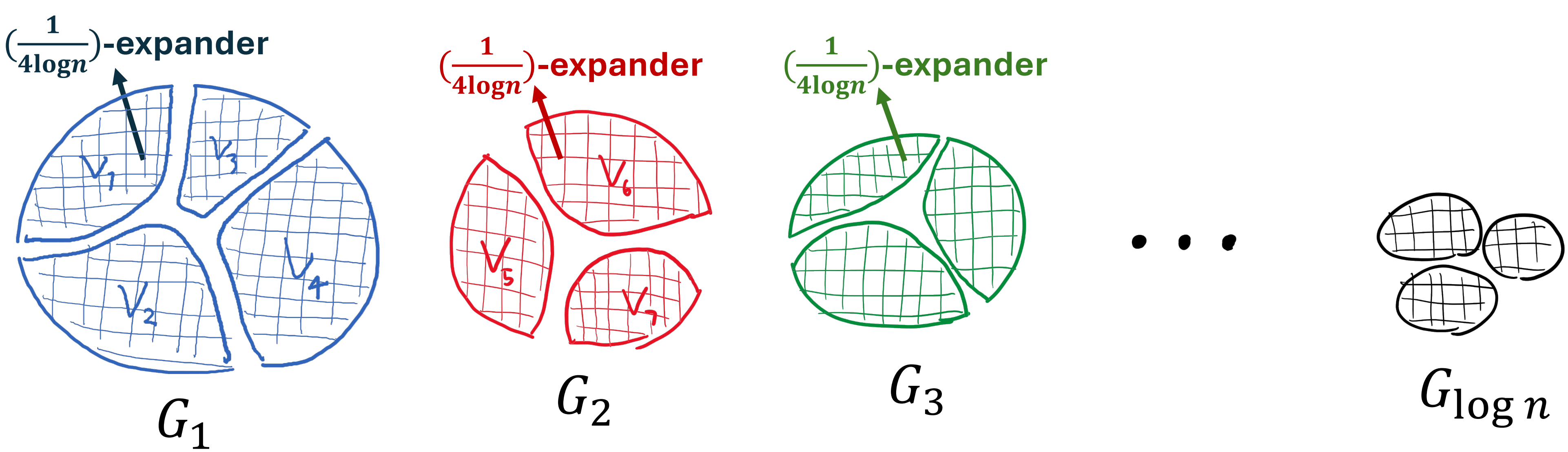
Repeated Expander Decomposition



Repeated Expander Decomposition



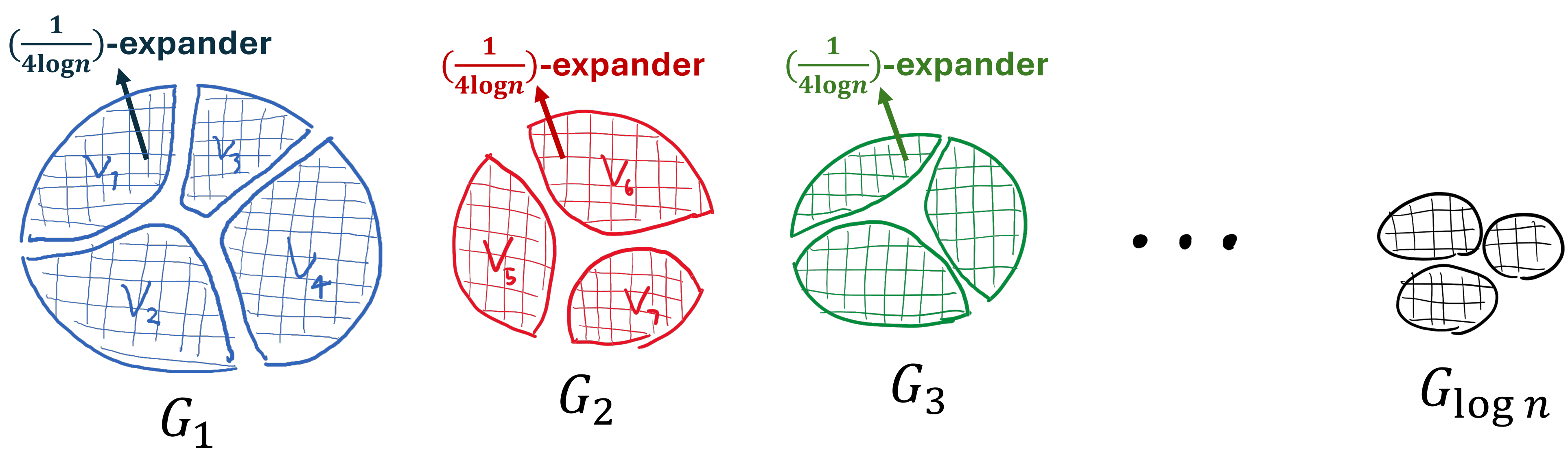
Repeated Expander Decomposition



Repeated Expander Decomposition

Theorem: Given $G = (V, E)$, can partition E

- Each part induces a $(\frac{1}{4\log})$-expander
- Each vertex is in $\leq \log n$ expanders



Part 6

Application of Expander Decomposition:
Edge Sparsifier

Edge Sparsifiers for Cuts

Input: graph $G = (V, E)$

Output: weighted graph $H = (V, E')$

- H has $\tilde{O}(n)$ weighted edges
- $w_G(S, V - S) \approx_{1+\epsilon} w_H(S, V - S)\ \forall S \subset V$

Sparsifier of ϕ-Expanders: **Degree-Sampling**

Linear-Time Algo: for each $e = (u, v)$

- Put edge e into H with prob $p_e = \min\{1, \frac{100 \log n}{\epsilon^2 \phi \min\{\deg_G u, \deg_G v\}}\}$
- Set weight of e to $1/p_e$

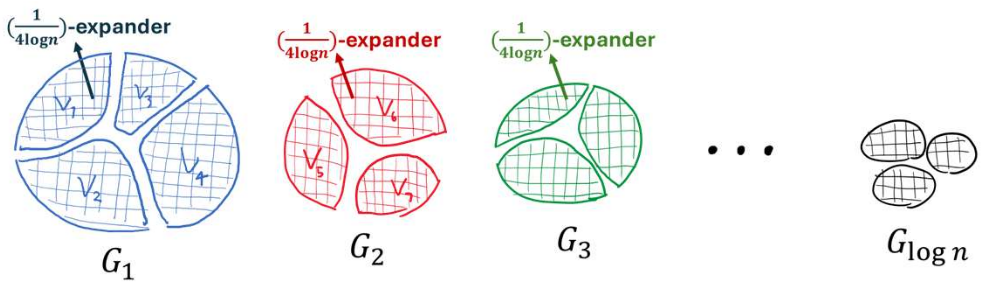
Correctness:

- $|E(H)| = \tilde{O}(n/\epsilon^2 \phi)$
  - Assign each edge to the lower degree endpoint.
  - Each vertex u is assigned $\leq \deg u$ edges, each of which is sampled with rate $\approx 1/\deg u$
- $(1 + \epsilon)$-approximation
  - This works as long as $p_{(u,v)} = \min\{1, \frac{100 \log n}{\epsilon^2 \lambda_{u,v}}\}$ [Fung Hariharan Harvey Panirahi]
  - We knew $\lambda_{u,v} \geq \phi \min\{\deg_G u, \deg_G v\}$ on ϕ-expander

Sparsifier on General Graphs

Algo:

1. $\{X_i\}_i \leftarrow$ repeated $(1/\,4\log n)$-expander decomposition of G
2. For each expander X_i, $\tilde{X}_i \leftarrow$degree-sampling(X_i)
3. Return $H = \cup_i \tilde{X}_i$



Size: $|E(H)| = \tilde{O}(n/\epsilon^2)$

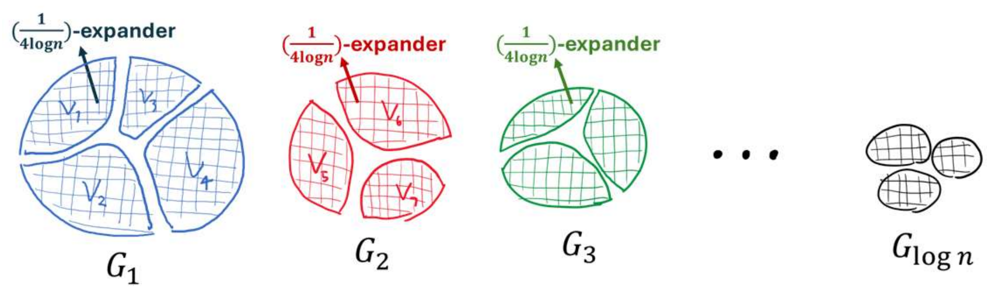
Approximation: union of sparsifiers is a sparsifier of the union

- Let $G = G_1 \cup G_2$. Let $\tilde{G}_1, \tilde{G}_2$ be α-sparsifier of G_1, G_2.
- Then, $\tilde{G} = \tilde{G}_1 \cup \tilde{G}_2$ is α-sparsifier of G

Sparsifier on General Graphs

Algo:

1. $\{X_i\}_i \leftarrow$ repeated $(1/\,4\log n)$-expander decomposition of G
2. For each expander X_i, $\tilde{X}_i \leftarrow$degree-sampling(X_i)
3. Return $H = \cup_i \tilde{X}_i$



Comment on this approach:

- First construction of "spectral sparsifiers" by [Spielman-Teng'04]
- Dynamic algorithm ⇒ ℓ_2-IPM for max flow in $\tilde{O}(m + n^{1.5})$ time

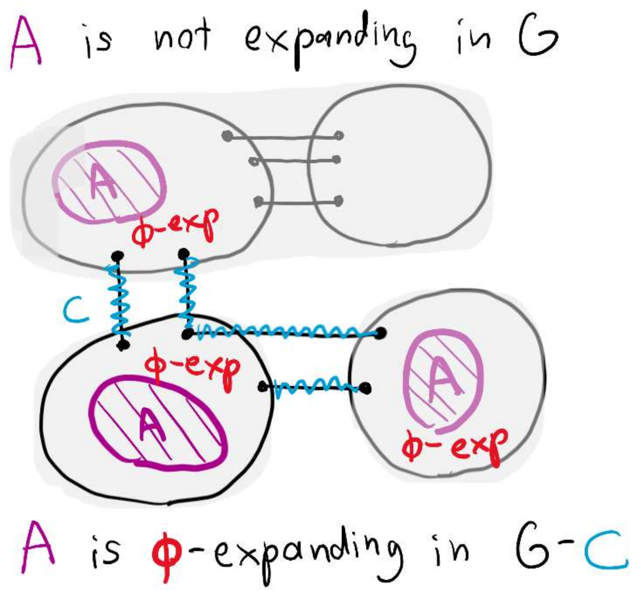
Part 7

Boundary-Linked Expander Decomposition

Recall: ϕ-expander decomposition of A in G

Theorem: Given $G = (V, E)$, A, ϕ, there exists $C \subseteq E$

- $|C| \leq (\phi \log n) \cdot |A|$
- A is ϕ-expanding in $G - C$



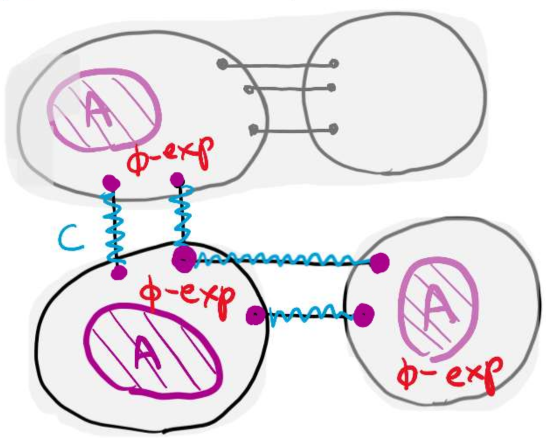
Will call C an **ϕ-ED** of A in G

Boundary-linked ϕ-expander decomposition of A in G

Theorem: Given $G = (V, E)$, A, $\phi \leq 1/4 \log n$, there exists $C \subseteq E$

- $|C| \leq (2\phi \log n) \cdot |A|$
- $A + \deg_C$ is ϕ-expanding in $G - C$

A is not expanding in G



$\deg_C$ + A is ϕ-expanding in G-C

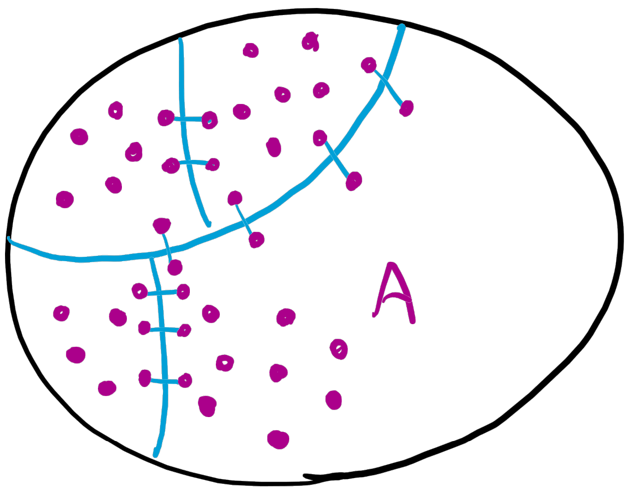
for each component U in $G - C$,
$A \cap U + \partial_G U$ is expanding in $G[U]$

Algorithm

- $C \leftarrow \emptyset$
- While $A' = A + \deg_C$ is not ϕ-expanding in $G - C$
  - So, $\exists$ ϕ-sparse cut $(S, U - S)$ in component U of $G - C$

    $$|E(S, U - S)| < \phi \min\{A'(S), A'(U - S)\}$$

  - $C \leftarrow C \cup E(S, U - S)$
- Return C



Analysis: After terminated

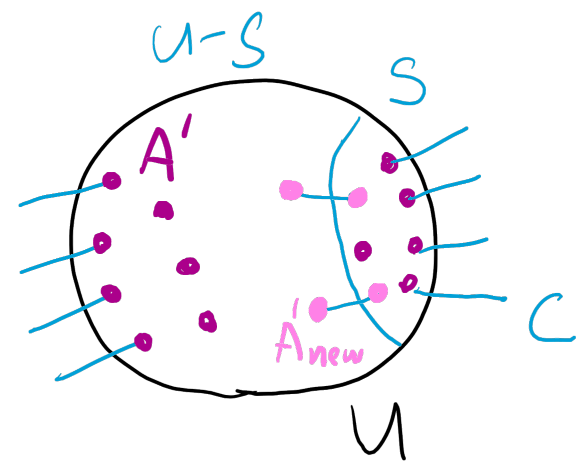
- A' is ϕ-expanding in $G - C$ ($A' \cap U$ is expanding in $G[U]$ $\forall U$)
- Remain to bound $|C|$

Bound $|C|$: Plan

- Initially, each A-vertex has $\$(2\phi A(v)\log n)$
- Without debt
  - Pay \$1 per edge in C $\Rightarrow |C| \leq 2\phi|A|\log n$
  - Maintain Invariant **"Each A'-vertex v has $\$(2\phi A'(v)\log|U_v|)$"**
    - U_v is the component in $G - C$ containing v"

Bound $|C|$: Payment scheme

When $C \leftarrow C \cup E(S, U - S)$, each A'-vertex in the smaller side pays $\$2\phi A'(v)$



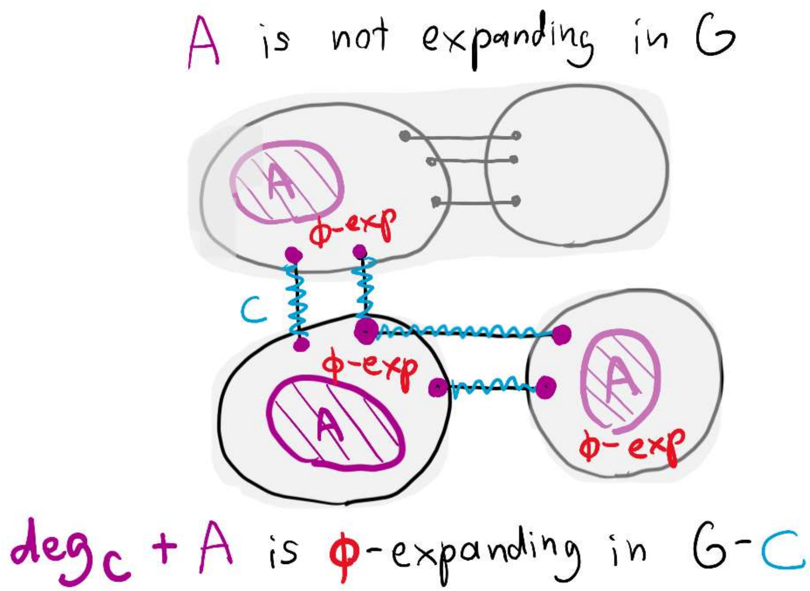
- Total Budget: $\$2\phi \min\{A'(S), A'(U - S)\}$
- Total Cost: $\$2E(S, U - S)$
  - $\$E(S, U - S)$ for new edges in C
  - $\$2\phi|A'_{new}|\log U$ to maintain invariant
    - $|A'_{new}| = 2|E(S, U - S)|$ as edges has two endpoints
    - $\phi \leq 1/4 \log n$
    - So, $\$2\phi|A'_{new}|\log U \leq \$E(S, U - S)$
- Cost $\leq$ Budget as S is ϕ-sparse ($E(S, U - S) \leq \phi \min\{A'(S), A'(U - S)\}$)

Observe: Invariant is maintained

Boundary-linked ϕ-expander decomposition of A in G

Theorem: Given $G = (V, E)$, A, $\phi \leq 1/4 \log n$, there exists $C \subseteq E$

- $|C| \leq (2\phi \log n) \cdot |A|$
- $A + \deg_C$ is ϕ-expanding in $G - C$



Part 8

Application of Boundary-Linked Expander Decomposition:

Vertex Sparsifiers

Vertex Sparsifiers: Informal

Given a huge graph G and a node weighting A.

Informal Goal:

- Compress G to size $\approx |A|$
- Preserve routability of all A-respecting demands

Vertex Sparsifiers

Given a huge graph G and a node weighting A.

Goal: find H s.t. for every A-respecting demand D

- D is routable in $G \Rightarrow D$ is routable in H
- D is routable in $H \Rightarrow D$ is routable in G with congestion $q = 4 \log n$
- $|E(H)| = O(|A|)$

Exercise: Preserve mincuts between *all subsets*.
for any $U \subseteq V$,

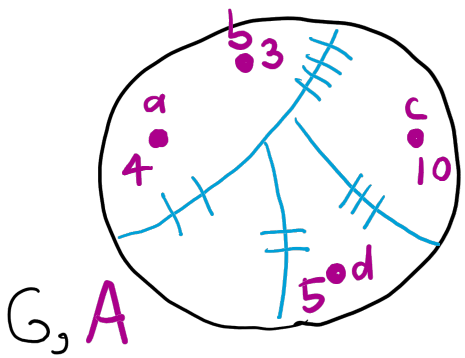
- For all $X, Y \subseteq U$,
  $\text{mincut}_G(X, Y) \leq \text{mincut}_H(X, Y) \leq q \cdot \text{mincut}_G(X, Y)$
- $|E(H)| = O\left(\deg_G(U) \log^2 n\right)$

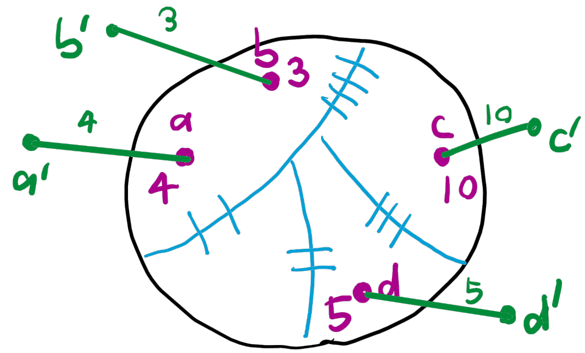
Construction [Chuzhoy'12]

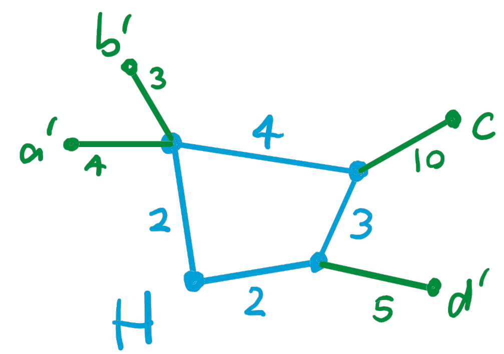
Theorem: Given G, A, $\phi \leq 1/4 \log n$, there is $C \subseteq E$
- $|C| \leq (2\phi \log n) \cdot |A|$
- $A + \deg_C$ is ϕ-expanding in $G - C$

1. Find C where $A + \deg_C$ is $(\phi = 1/4 \log n)$-expanding in $G - C$
2. For each A-vertex v, add edge (v, v') with capacity $A(v)$.
3. $H \leftarrow$ contract each component of $G - C$

Think:
v' represents v.





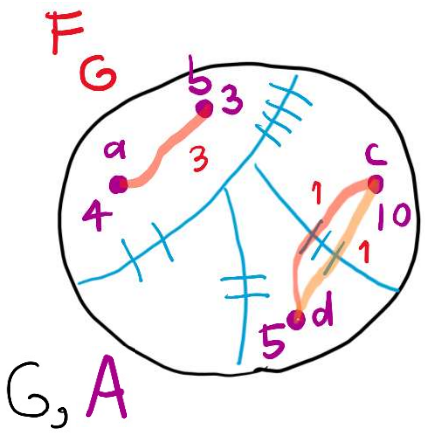


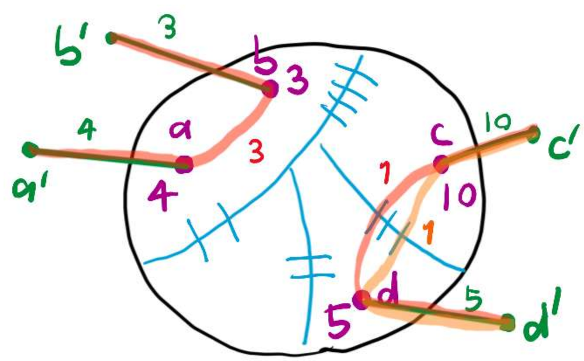
Size: $|E(H)| \leq |A| + |C| = O(|A|)$.

Next: show that H preserves routability

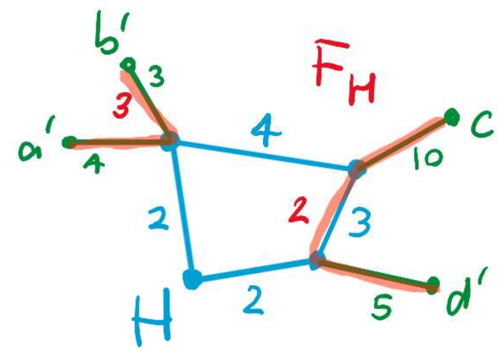
Routable in G $\Rightarrow$ Routable in H

- Let D be an A-respecting demand.
- Suppose F_G routes D in G with congestion 1
- **Goal**: Construct F_H routing D in H with congestion 1





D respects A
$\Rightarrow$ Dummy edges (v, v')
have congestion ≤ 1



Contraction never increases congestion.

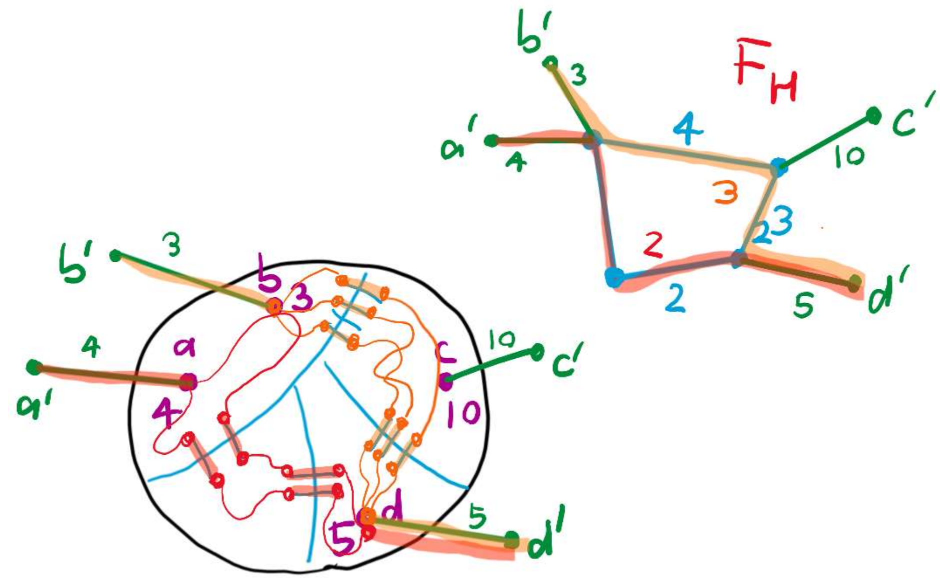
Routable in H ⇒ Routable in G with low congestion

- Let D be an A-respecting demand.
- Suppose F_H routes D in H with congestion 1
- **Goal**: Construct F_G routing D in G with congestion $q = 4 \log n$

For each component U in $G - C$

- F_H induces demand D_U
- D_U respects $(A + \deg_C) \cap U$
  - which is $(1/4 \log n)$-expanding in $G[U]$
- D_U is routable in $G[U]$ with congestion $4 \log n$

$F_G \leftarrow$ concatenate flow in G on each U



Vertex Sparsifiers

Given a huge graph G and a node weighting A.

Goal: find H s.t. for every A-respecting demand D

- D is routable in $G \Rightarrow D$ is routable in H
- D is routable in $H \Rightarrow D$ is routable in G with congestion $q = 4 \log n$
- $|E(H)| = O(|A|)$

Summary

What we learned

- Flow expansion ≈ Cut-expansion

- Easy algorithms on expanders
  - Approx max flow from degree

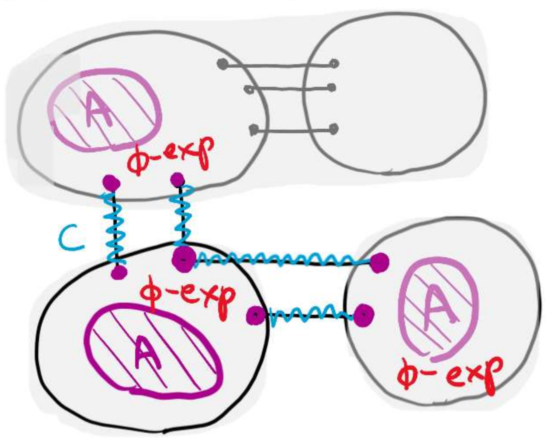
- Expander decomposition
  - Repeated expander decomposition
    - Application: edge sparsifiers for cuts
  - Boundary-linked expander decomposition
    - Application: vertex sparsifiers for flow

Boundary-linked ϕ-expander decomposition of A in G

Theorem: Given $G = (V, E)$, A, $\phi \leq 1/4 \log n$, there exists $C \subseteq E$

- $|C| \leq (2\phi \log n) \cdot |A|$
- $A + \deg_C$ is ϕ-expanding in $G - C$

A is not expanding in G



$\deg_C + A$ is ϕ-expanding in $G - C$

Repeated Expander Decomposition

Theorem: Given $G = (V, E)$, can partition E

- Each part induces a $(\frac{1}{4\log n})$-expander
- Each vertex is in $\leq \log n$ expanders