Workshop on Advances in Distributed Graph Algorithms (ADGA 2022)

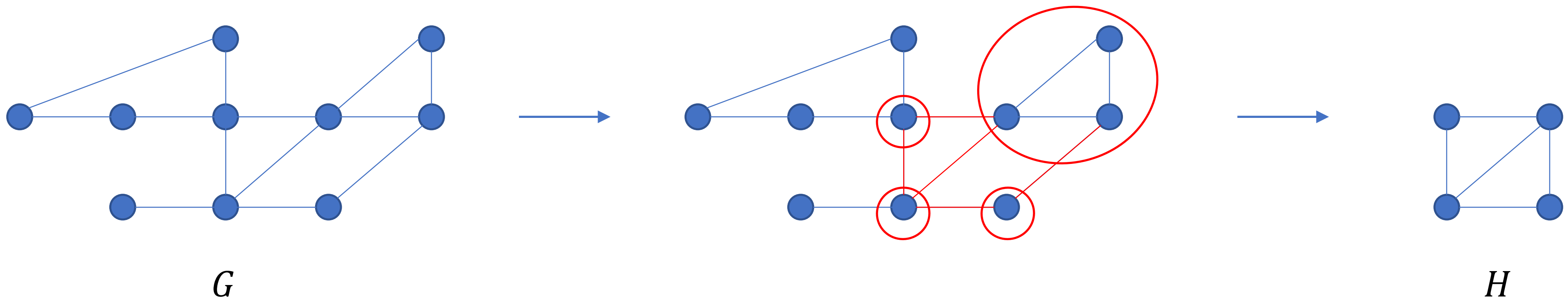
Distributed Graph Algorithms in Minor-Free Networks

Yi-Jun Chang

National University of Singapore

1. Minor-closed graph classes

- **Graph minor:**
  - H is a minor of G if H can be obtained from G by iteratively doing the following:
    1. Removing vertices.
    2. Removing edges.
    3. Contracting edges.



1. Minor-closed graph classes

- **Graph minor:**
  - H is a minor of G if H can be obtained from G by iteratively doing the following:
    1. Removing vertices.
    2. Removing edges.
    3. Contracting edges.

- Our focus: **minor-closed graph classes**.
  - A graph class is minor-closed if it is closed under taking minors.
    - (G is in the graph class) $\wedge$ (H is a minor of G) $\rightarrow$ (H is also in the graph class).
  - A graph class is minor-closed if it is closed under the above three operations.

1. Minor-closed graph classes

- The family of minor-closed graph classes include:
  - Forests.
  - Cactus graphs.
  - Planar graphs.
  - Outerplanar graphs.
  - Graphs of fixed genus g.
  - Graphs of treewidth at most k.
  - Graphs of pathwidth at most k.
  - …

1. Minor-closed graph classes

- **The graph minor theorem:** Any minor-closed graph class can be characterized by a finite list of excluded minors.

- **Examples:**
  - A graph is planar if and only if it is $\{K_{3,3}, K_5\}$-minor-free.
  - A graph is a forest if and only if it is K_3-minor-free.

- We may focus on the class of H-minor-free graphs.
  - For any minor-closed graph class that is not the set of all graphs, there exists H such that all graphs in this class are H-minor-free.

1. Minor-closed graph classes

- **Two key properties of H-minor free graphs:**
  - Arboricity $= O(1)$.
    - Uniformly sparse: Any subgraph has $m = O(n)$.
    - Lots of known algorithmic tools for bounded-arboricity graphs.
  - Closed under contraction.
    - Very relevant: Very often we consider a clustering and we want to work on the “cluster graph” which is the result of contracting each cluster into a vertex.

- We will see how they can be used in designing distributed algorithms.

2. Low-diameter decompositions

- A low-diameter decomposition removes a small fraction of edges so that each remaining connected component has small diameter.

- **Vertex version:**
  - Remove ϵ fraction of vertices.
  - Cluster the remaining vertices into non-adjacent subsets with diameter D.

- **Edge version:**
  - Remove ϵ fraction of edges.
  - Cluster the remaining vertices into non-adjacent subsets with diameter D.

2. Low-diameter decompositions

We will focus on the edge version with a strong diameter guarantee.

- **Vertex version:**
  - Remove ϵ fraction of vertices.
  - Cluster the remaining vertices into non-adjacent subsets with diameter D.
- **Edge version:**
  - Remove v fraction of edges.
  - Cluster the remaining vertices into non-adjacent subsets with diameter D.

Strong diameter: diameter of a cluster S is measured by the diameter of the subgraph $G[S]$ induced by S.

Weak diameter: diameter of a cluster S is measured by $\max_{u,v \in S} \operatorname{dist}(u, v)$, where the distance is measured in G.

2. Low-diameter decompositions

- Low-diameter decomposition is useful because it allows us to reduce from the general graph setting to the low-diameter graph setting.
- In particular, in the low-diameter setting, **brute-force information gathering** is possible in the LOCAL model.

2. Low-diameter decompositions

- **Applications:**
  - Network decompositions.
  - Expander decompositions and routing.
  - Densest subgraph detection.
  - $(1 \pm \epsilon)$-approximation for distributed covering and packing integer linear programs in $\text{poly}\left(\frac{1}{\epsilon}, \log n\right)$ rounds in LOCAL.

2. Low-diameter decompositions

Miller, Peng, and Xu, SPAA 2013

- A well-known **randomized** construction:
  - Cluster diameter: $O(\epsilon^{-1}\log n)$.
  - Round complexity: $O(\epsilon^{-1}\log n)$ in the CONGEST model.
  - The number of inter-cluster edges is at most $\epsilon|E|$ in expectation.

Ghaffari, Grunau, Haeupler, Ilchi, Rozhoň, SODA 2023

- The current best **deterministic** construction:
  - Cluster diameter: $\tilde{O}(\epsilon^{-1}\log n)$.
  - Round complexity: $\tilde{O}(\epsilon^{-1}\log^2 n)$ in the CONGEST model.
  - The number of inter-cluster edges is at most $\epsilon|E|$.

3. Ultra-fast algorithm in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- Low-diameter decompositions in H-minor-free networks:
  - Cluster diameter: $\epsilon^{-O(1)}$.
  - Round complexity: $\epsilon^{-O(1)} \cdot O(\log^* n)$ in the **LOCAL** model.
  - The number of inter-cluster edges is at most $\epsilon|E|$.

- We will show a proof sketch of this result.

3. Ultra-fast algorithm in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- Start with the trivial clustering:
  - Each vertex is a cluster.
- In each iteration:
  - Reduce the number of inter-cluster edges by a **constant factor**.
  - Growing the cluster diameter by a **constant factor**.
- $O(\log \epsilon^{-1})$ iterations suffice:
  - The number of inter-cluster edges $\leq \epsilon|E|$.
  - Cluster diameter $\leq \epsilon^{-O(1)}$.

3. Ultra-fast algorithm in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- Weighted by the multiplicity.
- Choose the **highest-weight** one.
- This requires sending large messages.

- **In each iteration:**
  - Take the cluster graph.
  - Let each cluster u chooses one of its neighboring cluster v and orient the edge $u \rightarrow v$.
  - This partitions the cluster graph into **rooted trees**.
  - Run an $O(\log^* n)$-round algorithm in each rooted tree to further divide the component into $O(1)$-diameter parts.

3. Ultra-fast algorithm in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- **Analysis:**
  - The cluster graph has bounded arboricity.
    - A constant fraction of the inter-cluster edges are oriented.
  - We can implement the final clustering step to ensure that a constant fraction of these inter-cluster edges will be within clusters at the end of this iteration.

↓

Indeed the number of inter-cluster edges is reduced by a **constant factor**.

4. Applications in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- Consider an optimization problem on graphs.
- We want to find a $(1 \pm \epsilon)$-approximate solution in the LOCAL model.

- **Idea:**
  - As long as the cost of ignoring inter-cluster edges is at most $\epsilon \cdot \mathrm{OPT}$, we may simply do a brute-force computation for each cluster.

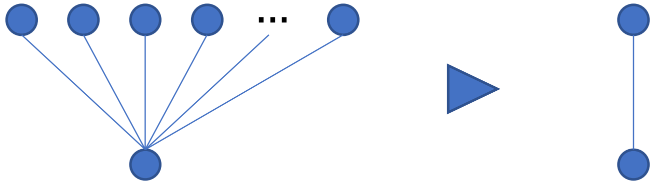
4. Applications in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- **Maximum independent set:**
  - A $(1-\epsilon)$-approximate solution of an ***H*-minor-free graph** can be computed in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds deterministically in the LOCAL model.

- **Proof:**
  - For bounded-arboricity graphs, $\mathrm{OPT} = \Theta(n)$.
  - We can afford to ignore all inter-cluster edges.

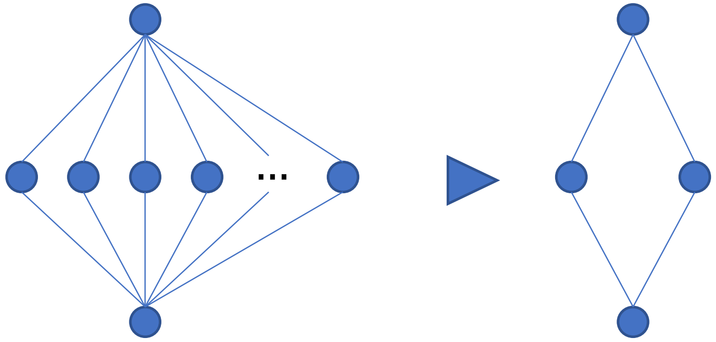
4. Applications in minor-free networks



Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- **Maximum matching:**
  - A $(1-\epsilon)$-approximate solution of a **planar graph** can be computed in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds deterministically in the LOCAL model.

- **Proof:**
  - We do not have $\mathrm{OPT} = \Theta(n)$ in general, due to two structures.
  - We may do a preprocessing to remove these structures.
  - After that, $\mathrm{OPT} = \Theta(n)$.



4. Applications in minor-free networks

Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- **Minimum dominating set:**
  - A $(1+\epsilon)$-approximate solution of a **planar graph** can be computed in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds deterministically in the LOCAL model.

- **Proof:**
  - We do not have $\mathrm{OPT} = \Theta(n)$ in general.
  - First compute an $O(1)$-approximate solution D.
  - Each vertex $v \in V \setminus D$ joins the cluster of any $u \in N(v) \cap D$.
  - Compute a low-diameter decomposition of the cluster graph.
  - Now the number of inter-cluster edges is at most $\epsilon \cdot \mathrm{OPT}$.

5. Distributed property testing

- **Distributed property testing:**
  - If G has property $\mathcal{P}$, then all vertices output **accept**.
  - If G is **ϵ-far** from having property $\mathcal{P}$, then at least one vertex outputs **reject**.

(To obtain property $\mathcal{P}$, we need to insert or delete at least $\epsilon|E|$ edges.)

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

(Deterministic)

- **Algorithm:**
  - Compute a low-diameter decomposition.
  - Each cluster locally decide if it has the property $\mathcal{P}$.

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

- **Algorithm:**
  - Compute a low-diameter decomposition.
  - Each cluster locally decide if it has the property $\mathcal{P}$.

A subtle issue:

- The bound $\epsilon|E|$ on the number of inter-cluster edges **is not guaranteed** if the underlying graph does not have property $\mathcal{P}$.

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

- **Algorithm:**
  - Compute a low-diameter decomposition.
  - Each cluster locally decide if it has the property $\mathcal{P}$.

Solution:

- To ensure that the bound $\epsilon|E|$ holds, all we need is that the cluster graph has small arboricity.
  - We can run an $\boldsymbol{O(\log n)}$**-round** algorithm in each iteration to check whether the arboricity bound holds.
  - If the arboricity bound does not hold, then some vertex will detect it and output **reject**.

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

- **Algorithm:**
  - Compute a low-diameter decomposition.
  - Each cluster locally decide if it has the property $\mathcal{P}$.

- If **accept** for all clusters:
  - The union of all clusters still has the property $\mathcal{P}$.
  - The original graph is at most ϵ-far from having the property $\mathcal{P}$.

$\mathcal{P}$ is closed under taking disjoint union.

Number of inter-cluster edges is at most $\epsilon|E|$.

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

- **Algorithm:**
  - Compute a low-diameter decomposition.
  - Each cluster locally decide if it has the property $\mathcal{P}$.

$\mathcal{P}$ is closed under taking minor.

- If **reject** for at least one cluster:
  - The original graph does not have property $\mathcal{P}$.

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

- **Lower bound:**
  - The $O(\log n)$ factor is necessary.

Levi, Medina, and Ron, PODC 2018

5. Distributed property testing

- Property testing of any **minor-closed property** that is **closed under disjoint union** can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds in LOCAL.

- The "closed under disjoint union" condition **cannot be removed**:
  - There is a minor-closed graph property that is not closed under disjoint union requiring $\Omega(n)$ rounds to test.

Chang and Su, PODC 2022

6. Extension to CONGEST

- **Question:** Can we extend these LOCAL algorithms to CONGEST?
  - Approximate maximum matching
  - Approximate maximum independent set
  - Approximate minimum dominating set
  - Property testing a minor-closed property that is closed under disjoint union
  - …

6. Extension to CONGEST

- **Two barriers:**
  1. Need an efficient CONGEST algorithm for the low-diameter decomposition.
     - Can just use the existing CONGEST ones, although they are less efficient.

  2. Need to replace the “brute-force information gathering” part with an efficient CONGEST algorithm.
     - Seems to require a CONGEST algorithm that is efficient for small-diameter networks.

6. Extension to CONGEST

- **Planarity testing** can be done in $O(D \log n)$ rounds in **CONGEST**.

Ghaffari and Haeupler, PODC 2016

Low-diameter decomposition of planar graphs in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds with high probability in **CONGEST**.

Levi, Medina, and Ron, PODC 2018

- Property testing of planarity can be done in $\epsilon^{-O(1)} \cdot O(\log n)$ rounds with high probability in the **CONGEST** model.

Levi, Medina, and Ron, PODC 2018

6. Extension to CONGEST

Levi, Medina, and Ron, PODC 2018

- For the special case of property testing of planarity, it is possible to overcome these barriers to obtain an efficient CONGEST algorithm.

Matching the round complexity in the LOCAL model.

- What about other problems?

- **How can we narrow this gap between LOCAL and CONGEST?**

7. Expander decompositions

- To answer that question, a natural approach is to consider **expander decomposition**, which can be seen as an analogue of low-diameter decomposition for the CONGEST model.

7. Expander decompositions

Consider a graph $G = (V, E)$.

Volume of a vertex set S:

- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.

Conductance of a cut $(S, V \setminus S)$:

- $\Phi(S) = \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}$, where $E(A, B) = \{ \{u, v\} \in E \mid u \in A \text{ and } v \in B \}$.

Conductance of a graph G:

- $\Phi(G) = \min_{S \subseteq V \text{ s.t. } S \neq V \text{ and } S \neq \emptyset} \Phi(S)$.

7. Expander decompositions

This allows us to reduce from general graphs to high-conductance graphs.

Expander decompositions:

For every graph, it is possible to remove a small ϵ fraction of the edges so that each remaining connected component has high conductance ϕ.

7. Expander decompositions

This allows us to reduce from general graphs to high-conductance graphs.

▽

Expander decompositions:

For every graph, it is possible to remove a small ϵ fraction of the edges so that each remaining connected component has high conductance ϕ.

Expander routing:

In a high-conductance graph, each vertex v can very quickly exchange messages with $\deg(v)$ arbitrary vertices, not just the neighbors of v.

△

This is a useful communication primitive for designing algorithms on high-conductance networks.

7. Expander decompositions

- **Expander decomposition:**
  - Randomized:
    - Conductance: $\phi = \frac{1}{\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)}$.
    - Round complexity: $\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)$.
  - Deterministic:
    - Conductance: $\phi = \frac{1}{n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\epsilon}\right)}$.
    - Round complexity: $n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\epsilon}\right)$.
- **Expander routing:**
  - Randomized and deterministic:
    - Round complexity: $n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\phi}\right)$.

7. Expander decompositions

- Applications of expander decomposition in the CONGEST model:
  - It has become a standard technique in distributed subgraph finding.

    Chang, Pettie, and Zhang, SODA 2019

    Chang and Saranurak, PODC 2019

    Eden, Fiat, Fischer, Kuhn, and Oshman, DISC 2019

    ⋮

    Censor-Hillel, Leitersdorf, and Vulakh, PODC 2022
  - It has been applied to exact minimum cut computation.

    Daga, Henzinger, Nanongkai, and Saranurak, STOC 2019

7. Expander decompositions

- The usage of expander decompositions is still **limited** in CONGEST, as it does not allow us to do brute-force information gathering in each cluster.

- **Can we bypass this barrier?**

8. Separator theorems

- Let's consider planar graphs.

- **Planar separator theorem:**
  - For any planar graph, we can remove $O(\sqrt{n})$ **vertices** to partition the graph into disjoint subgraphs with at most $\frac{2n}{3}$ vertices.

8. Separator theorems

- Let's consider planar graphs.

- **Planar separator theorem:**
  - For any planar graph, we can remove $O(\sqrt{n})$ **vertices** to partition the graph into disjoint subgraphs with at most $\frac{2n}{3}$ vertices.

- **A slightly less-known result:**
  - For any planar graph, we can remove $O\left(\sqrt{\Delta n}\right)$ **edges** to partition the graph into disjoint subgraphs with at most $\frac{2n}{3}$ vertices.

Edge separator theorem

8. Separator theorems

- **Key observation:**

G is a planar graph with conductance at least ϕ.

$\bigtriangledown$

Since G is a planar, G has an edge separator of size $O\left(\sqrt{\Delta n}\right)$.
Since the conductance of G is at least ϕ, we have $\sqrt{\Delta n} = \Omega(\phi n)$.

G has maximum degree $\Delta = \Omega(\phi^2 n)$.

8. Separator theorems

- **Key observation:**

G is a planar graph with conductance at least ϕ.

$\bigtriangledown$

Since G is a planar, G has an edge separator of size $O(\sqrt{\Delta n})$.
Since the conductance of G is at least ϕ, we have $\sqrt{\Delta n} = \Omega(\phi n)$.

G has maximum degree $\Delta = \Omega(\phi^2 n)$.

$\bigtriangledown$

Using random walks, the entire topology of G can be gathered to a vertex efficiently, in $\mathrm{poly}(\phi^{-1}, \log n)$ rounds, with high probability.

8. Separator theorems

- **Key observation:**

G is a planar graph with conductance at least ϕ.

▽

Since G is a planar, G has an edge separator of size $O\left(\sqrt{\Delta n}\right)$.
Since the conductance of G is at least ϕ, we have $\sqrt{\Delta n} = \Omega(\phi n)$.

G has maximum degree $\Delta = \Omega(\phi^2 n)$.

▽

This can also be done in deterministic $n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\phi}\right)$ rounds.

Using random walks, the entire topology of G can be gathered to a vertex efficiently, in $\mathrm{poly}(\phi^{-1}, \log n)$ rounds, with high probability.

8. Separator theorems

- What is the broadest natural graph class that allows each cluster of an expander decomposition to have small edge separator?

- For any **bounded-genus graph**, we can remove $O\left(\sqrt{\Delta n}\right)$ edges to partition the graph into disjoint subgraphs with at most $\frac{2n}{3}$ vertices.

Sykora and Vrto, Theoretical Computer Science, 1993

8. Separator theorems

- What is the broadest natural graph class that allows each cluster of an expander decomposition to have small edge separator?

- For any **bounded-genus graph**, we can remove $O\left(\sqrt{\Delta n}\right)$ edges to partition the graph into disjoint subgraphs with at most $\frac{2n}{3}$ vertices.

Sykora and Vrto, Theoretical Computer Science, 1993

- For any $\boldsymbol{H}$**-minor-free graph,** we can remove $O\left(\sqrt{\Delta n}\right)$ edges to partition the graph into disjoint subgraphs with at most $\frac{2n}{3}$ vertices.

Chang and Su, PODC 2022

9. Applications of expander decompositions

- Most of the previously discussed LOCAL algorithms can be transformed into CONGEST algorithms.
  - **Randomized:** $\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)$ rounds.
  - **Deterministic:** $n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\epsilon}\right)$ rounds.

Chang and Su, PODC 2022

9. Applications of expander decompositions

- The list of problems include:
  - $(1-\epsilon)$-approximate maximum independent set on H-minor-free graphs.
  - $(1-\epsilon)$-approximate maximum matching on planar graphs.
  - Property testing any minor-closed graph property that is closed under disjoint union.
  - …

Chang and Su, PODC 2022

9. Applications of expander decompositions

- Our approach does not seem to apply to $(1 + \epsilon)$-approximate minimum dominating set.
  - The reason is that in the dominating set algorithm the low-diameter decomposition is applied to the cluster graph, not the original graph.

10. Faster expander decomposition algorithm

Question: Can we further improve these bounds by utilizing the structural properties of H-minor-free graphs?

- **Expander decomposition:**
  - Randomized:
    - Conductance: $\phi = \frac{1}{\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)}$.
    - Round complexity: $\mathrm{poly}\left(\log n\,, \frac{1}{\epsilon}\right)$.
  - Deterministic:
    - Conductance: $\phi = \frac{1}{n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\epsilon}\right)}$.
    - Round complexity: $n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\epsilon}\right)$.
- **Expander routing:**
  - Randomized and deterministic:
    - Round complexity: $n^{o(1)} \cdot \mathrm{poly}\left(\frac{1}{\phi}\right)$.

10. Faster expander decomposition algorithm

Claim: The following improved bounds can be achieved for H-minor-free graphs.

- **Expander decomposition:**
  - Deterministic:
    - Conductance: $\phi = \frac{1}{\text{poly}\left(\log n, \frac{1}{\epsilon}\right)}$.
    - Round complexity: $\text{poly}\left(\log n, \frac{1}{\epsilon}\right)$.
- **Expander routing:**
  - Deterministic:
    - Round complexity: $\text{poly}\left(\log n, \frac{1}{\phi}\right)$.

10. Faster expander decomposition algorithm

Using an existing deterministic CONGEST algorithm on general graphs

- High-level idea:
  - Find a **low-diameter decomposition**. ◁ $\text{poly}\left(\log n, \frac{1}{\epsilon}\right)$ rounds
    - Remove all inter-cluster edges.
  - For each cluster, find a **balanced sparse cut**. ◁ ?
    - Remove all cut edges.
  - Recurse on each remaining connected component.

10. Faster expander decomposition algorithm

- Balanced sparse cut computation:
  - Partition the vertex set into $\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)$ connected parts.
    - Each part has roughly the same number of incident edges.
    - This partition can be computed by processing any BFS tree in a bottom-up manner.
  - Consider the cluster graph.
    - Each part is contracted into a vertex.
  - Compute a **balanced vertex separator** for the cluster graph.
    - Use brute-force information gathering.
    - Remove all the edges incident to the parts in the separator.

The cluster graph is still H-minor-free.

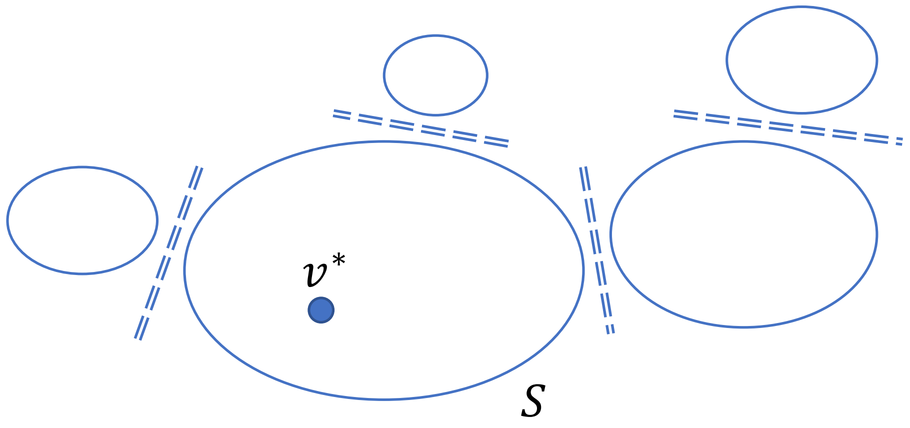
This costs $\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)$ rounds.

10. Faster expander decomposition algorithm

- The guarantee that each part has roughly the same number of incident edges works only if there is no **high-degree vertex**.
  - Need to switch to a different approach if a high-degree vertex exists.

10. Faster expander decomposition algorithm

- Let v^* be a high-degree vertex.
- **High-level idea:**
  - We try to let v^* learn as much as possible about the graph topology.
  - If the learning speed is too slow, then there must be a **sparse cut**.
  - We will identify the sparse cut and remove all the cut edges.
  - In the end, v^* can learn all information about the component S that it belongs to.



- We will recurse on each component.
- For S, we may use brute-force computation.

10. Faster expander decomposition algorithm

- For information gathering, we use a **load balancing** algorithm on high-conductance bounded-degree graphs.

Ghosh, Leighton, Maggs, Muthukrishnan, Plaxton, Rajaraman, Richa, Tarjan, and Zuckerman, SIAM Journal on Computing 1999

- We can simulate a bounded-degree graph by letting v simulates $\mathrm{deg}(v)$ vertices.

10. Faster expander decomposition algorithm

Ghosh, Leighton, Maggs, Muthukrishnan, Plaxton, Rajaraman, Richa, Tarjan, and Zuckerman, SIAM Journal on Computing 1999

- The way this algorithm works is that whenever $\mathrm{load}(u) - \mathrm{load}(v)$ is too high for some edge $\{u, v\}$, then we send some items from u to v.
  - If the underlying graph has conductance ϕ, then after $\mathrm{poly}\left(\log n, \frac{1}{\phi}\right)$ rounds each vertex will roughly have the same load.
  - Otherwise, a sparse cut can be found.

10. Faster expander decomposition algorithm

Ghosh, Leighton, Maggs, Muthukrishnan, Plaxton, Rajaraman, Richa, Tarjan, and Zuckerman, SIAM Journal on Computing 1999

- We omit the technical details of how this load balancing algorithm is used to implement the high-level idea discussed earlier.

- This also allows us to solve **expander routing** in $\mathrm{poly}\left(\log n, \frac{1}{\phi}\right)$ rounds deterministically in H-minor-free graphs with conductance ϕ.

10. Faster expander decomposition algorithm

- **Expander decomposition:**
  - Deterministic:
    - Conductance: $\phi = \frac{1}{\text{poly}\left(\log n, \frac{1}{\epsilon}\right)}$.
    - Round complexity: $\text{poly}\left(\log n, \frac{1}{\epsilon}\right)$.

- **Expander routing:**
  - Deterministic:
    - Round complexity: $\text{poly}\left(\log n, \frac{1}{\phi}\right)$.

Corollary: All $n^{o(1)} \cdot \text{poly}\left(\frac{1}{\epsilon}\right)$-round deterministic algorithms for H-minor-free graphs discussed earlier can be implemented to run in $\text{poly}\left(\log n, \frac{1}{\epsilon}\right)$ rounds deterministically.

11. Ultra-fast expander decomposition

Question: Is $\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)$ the best we can hope for?

- Finding an expander decomposition in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds for H-minor-free graphs in CONGEST does not seem to contradict any known lower bounds.

11. Ultra-fast expander decomposition

Claim: $\epsilon^{-O(1)} \cdot O(\log^* n)$ can be achieved for **bounded-degree** graphs.

- **Idea:**
  - We want to turn the LOCAL $\epsilon^{-O(1)} \cdot O(\log^* n)$-round low-diameter decomposition algorithm into a CONGEST one.

11. Ultra-fast expander decomposition

- The main reason that the LOCAL $\epsilon^{-O(1)} \cdot O(\log^* n)$-round low-diameter decomposition algorithm needs **large messages**:
  - Each cluster A needs to identify a neighboring cluster B such that the number of inter-cluster edges between A and B is maximized.

- **Observation:**
  - If each cluster has high conductance, then this task can be solved with small messages.

11. Ultra-fast expander decomposition

- We will **modify** the LOCAL $\epsilon^{-O(1)} \cdot O(\log^* n)$-round low-diameter decomposition algorithm as follows:
  - For each cluster, run the $\mathrm{poly}\left(\log n, \frac{1}{\epsilon}\right)$-round deterministic expander decomposition algorithm.
  - After that, each cluster has enough message processing capability.

11. Ultra-fast expander decomposition

Round complexity:

- In a low-diameter decomposition, the size of a cluster is at most $\Delta^{\mathrm{poly}\left(\frac{1}{\epsilon}\right)}$.
- For $n' = \Delta^{\mathrm{poly}\left(\frac{1}{\epsilon}\right)}$, we have:
  - $\mathrm{poly}\left(\log n', \frac{1}{\epsilon}\right) = \mathrm{poly}\left(\log \Delta, \frac{1}{\epsilon}\right)$.
- The overall round complexity of our expander decomposition algorithm is:
  - $\mathrm{poly}\left(\log \Delta, \ \frac{1}{\epsilon}\right) \cdot O(\log^* n)$.

For **bounded-degree graphs**, this is $\epsilon^{-O(1)} \cdot O(\log^* n)$.

11. Ultra-fast expander decomposition

- **Expander decomposition:**
  - Deterministic:
    - Conductance: $\phi = \frac{1}{\text{poly}\left(\log\Delta, \frac{1}{\epsilon}\right)}$.
    - Round complexity: $\text{poly}\left(\log\Delta\,,\ \frac{1}{\epsilon}\right) \cdot O(\log^* n)$.

Question: Are there any implications beyond **bounded-degree** graphs?

12. Bounded degree sparsifiers

"Local Algorithms for Bounded Degree Sparsifiers in Sparse Graphs" by Solomon, ITCS 2018

- There exist **bounded degree sparsifiers** which allow us to reduce from bounded-arboricity graphs to the bounded-degree graphs.

12. Bounded degree sparsifiers

"Local Algorithms for Bounded Degree Sparsifiers in Sparse Graphs" by Solomon, ITCS 2018

- There exist **bounded degree sparsifiers** which allow us to reduce from bounded-arboricity graphs to the bounded-degree graphs.

Graphs with arboricity α	One-round reduction	The same problem in graphs with maximum degree Δ
$(1-\epsilon)$-approximate maximum matching	→	$\Delta = O(\alpha/\epsilon)$
$(1-\epsilon)$-approximate maximum independent set	→	$\Delta = O(\alpha^2/\epsilon)$
$(1+\epsilon)$-approximate vertex cover	→	$\Delta = O(\alpha/\epsilon)$

12. Bounded degree sparsifiers

"Local Algorithms for Bounded Degree Sparsifiers in Sparse Graphs" by Solomon, ITCS 2018

Graphs with arboricity α	One-round reduction	The same problem in graphs with maximum degree Δ
$(1-\epsilon)$-approximate maximum matching	→	$\Delta = O(\alpha/\epsilon)$
$(1-\epsilon)$-approximate maximum independent set	→	$\Delta = O(\alpha^2/\epsilon)$
$(1+\epsilon)$-approximate vertex cover	→	$\Delta = O(\alpha/\epsilon)$

Corollary: All these problems can be solved in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds deterministically in H-minor-free graphs in CONGEST.

Expander decomposition

In $\mathrm{poly}\left(\log \Delta, \frac{1}{\epsilon}\right) \cdot O(\log^* n)$ rounds

13. Conclusion and open questions

Theorem: $(1-\epsilon)$-approximate maximum matching and maximum independent set can be solved in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds deterministically in H-minor-free graphs in CONGEST.

- The round complexity **matches** the algorithms in LOCAL model:

  Czygrinow, Hanckowiak, and Wawrzyniak, DISC 2008

- **Techniques:**
  - Improved deterministic expander decomposition.
  - Bounded-degree sparsifier. Solomon, ITCS 2018

13. Conclusion and open questions

$\boldsymbol{H}$-minor-free graphs:

- Expander decomposition in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds in CONGEST?
- What other problems can be solved in $\epsilon^{-O(1)} \cdot O(\log^* n)$ rounds in CONGEST?
- What other problems admit a bounded-degree sparsifier?

13. Conclusion and open questions

H-minor-free graphs:

- Other non-trivial applications of our approach?

- The complexity of $(1+\epsilon)$-approximate minimum dominating set in CONGEST?

13. Conclusion and open questions

- Characterization of efficiently testable minor-closed graph properties in CONGEST?

- **Upper bound:**
  - Any minor-closed graph property $\mathcal{P}$ **closed under disjoint union** admits an efficient property testing algorithm:
    - Deterministic $\text{poly}\left(\log n, \frac{1}{\epsilon}\right)$ rounds.

Is this the right characterization?

13. Conclusion and open questions

- Characterization of efficiently testable minor-closed graph properties in CONGEST?

- **Upper bound:**
  - Any minor-closed graph property $\mathcal{P}$ **closed under disjoint union** admits an efficient property testing algorithm:
    - Deterministic $\text{poly}\left(\log n, \frac{1}{\epsilon}\right)$ rounds.

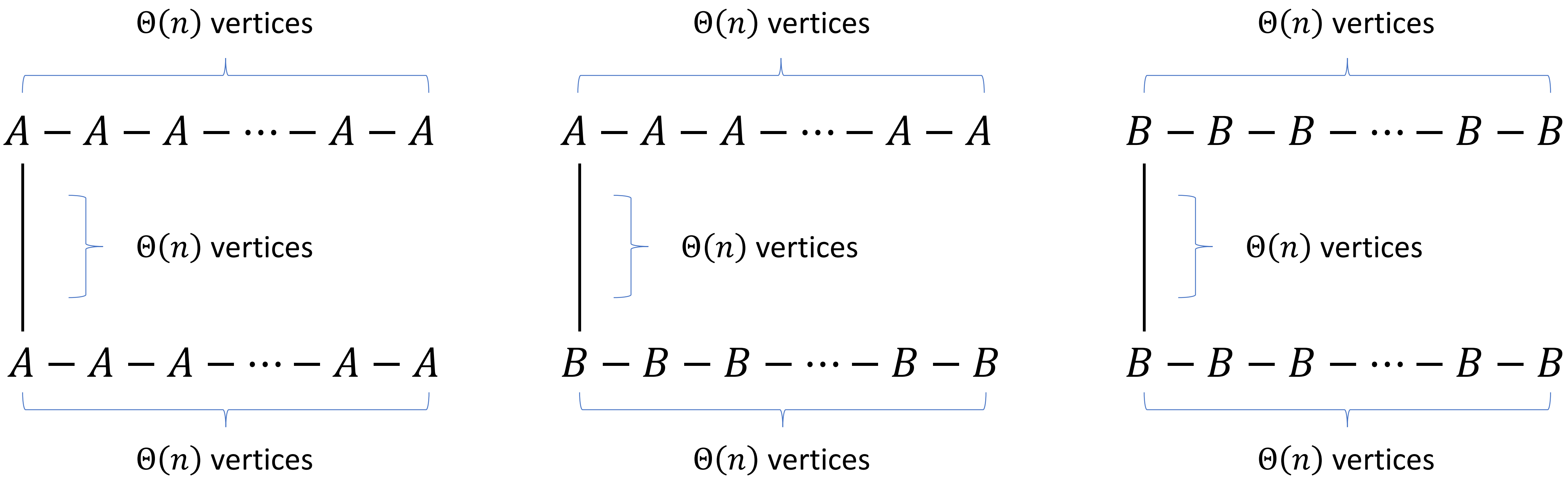
Is this the right characterization?

$\mathcal{P} = \{\text{graphs that can be embedded on a torus}\}$.
- $\mathcal{P}$ is not closed under disjoint union.
- Does $\mathcal{P}$ admit an efficient property testing algorithm?

13. Conclusion and open questions

$H = A \cup B$ for some **suitable** choices of A and B.

- **Proof sketch** of an $\Omega(n)$ lower bound for a minor-closed graph property:



***H*-minor-free** | **Very far from *H*-minor-free** | ***H*-minor-free**

13. Conclusion and open questions

- This $\Omega(n)$ lower bound only applies to **some of** the minor-closed graph properties that are not closed under disjoint union.

- In particular, the lower bound **does not** apply to
  - $\mathcal{P} = \{\text{graphs that can be embedded on a torus}\}$.

13. Conclusion and open questions

- This $\Omega(n)$ lower bound only applies to **some of** the minor-closed graph properties that are not closed under disjoint union.

- In particular, the lower bound **does not** apply to
  - $\mathcal{P} = \{\text{graphs that can be embedded on a torus}\}$.

Thank you!