Worst-Case to Expander-Case Reductions: Derandomized and Generalized

Amir Abboud and *Nathan Wallheimer*

WEIZMANN INSTITUTE OF SCIENCE

European Research Council
Established by the European Commission

Question 1: Are expanders worst-case instances of our problem?

Motivation?

Expanders: highly connected, random-like graphs. Expanders are one of the most important graph families in computer science.

Formal definition: Let $vol(S) = \sum_{v \in S} \deg(v)$

- $\forall S \subseteq V : vol(S) > 0$
- $\forall S \subseteq V : \frac{e(S, V \setminus S)}{\min(vol(S), vol(V \setminus S))} \geq \phi = \Omega(1)$

How to prove lower bounds for expanders?

- Direct lower bounds - difficult even for non-expanders.
- Conditional lower bounds - does not work for any problem, e.g., problems like Maximum Matching that don't have conditional lower bounds.
- **Our focus:** self-reductions to expanders.

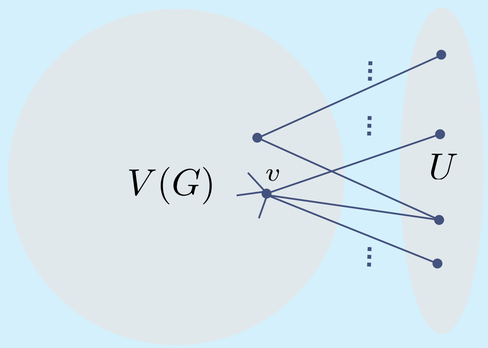
Previous Work (AW23): Direct Worst-case to Expander-case Reductions

Given input G, output G_{exp} such that:

- G_{exp} is an expander.
- The solution to G can be computed in linear time from the solution to G_{exp}.
- The blowup in the size of G_{exp} is linear in the size of G.

The core gadget (AW23)

- Add an expansion layer U of n vertices.
- From every $v \in V(G)$, add $\deg(v)$ random neighbors in U.



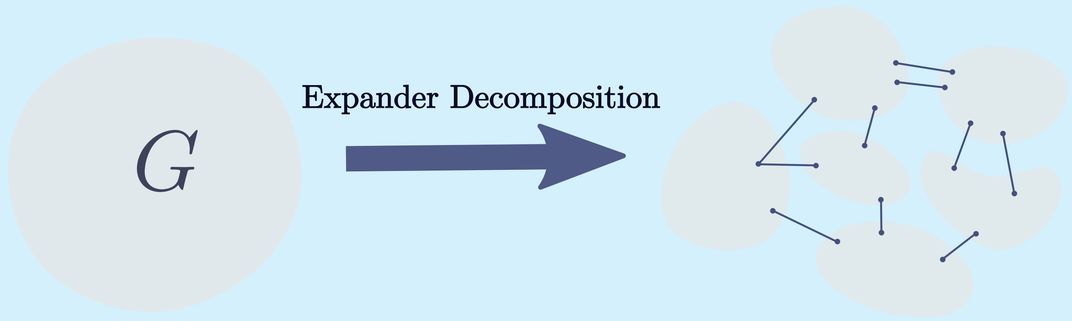
The resulting graph is an expander with high proability.

Limitations to AW23

- **Randomized** - does not imply hardness on expanders for deterministic algorithms. Does not rule out derandomization via expander decompositions.
- **Static** - does not address other models such as the fully-dynamic model, where edges are inserted and deleted from the graph.

Additional motivation

The expander decomposition method

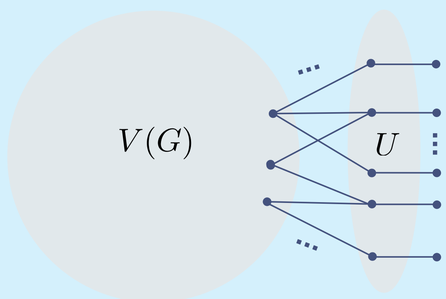


- Max-flow in $m^{1+o(1)}$ time.
- **Derandomized** global min-cut in $m^{1+o(1)}$ time.
- **Dynamic** minimum spanning tree in $n^{o(1)}$ update.

The basis for many recent breakthroughs!

Direct worst-case to expander-case reductions show that the expander decomposition method is essentially useless!

Example: Maximum Matching (AW23)



Apply the core gadget and then add pendant vertices to U.

$MM(G_{exp}) = MM(G) + n$

- Expanders are worst-case instances of Maximum Matching.
- Expander decomposition is useless against Maximum Matching.

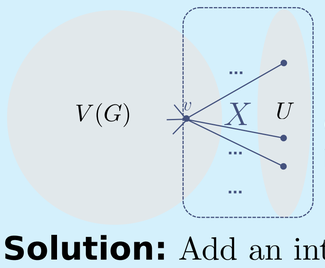
Question 2: Can we derandomize the best algorithms for our problem via expander decomposition?

This Work: Derandomized, Dynamic Core Gadget

Derandomized Core Gadget

Attempt: employ a *fully explicit* expander construction between V and U.

For any sufficiently large n and $d \geq 3$, can construct a d-regular expander X.

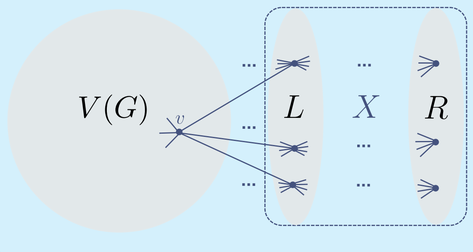


Which degree d should we pick?

$d = \Theta(n)$ — Good expansion but large blowup.

$d = \Theta(2m/n)$ — Not enough to expand cuts with many internal edges. Works only if the graph is regular.

Solution: Add an intermediate level for degree balancing.



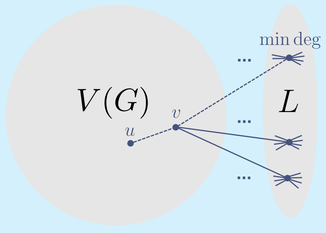
Connect every $v \in V(G)$ to $\deg(v)$ neighbors in L using a deterministic load balancing algorithm. Pick $d = \Theta(2m/n)$.

Fully-Dynamic Core Gadget

Goal: maintain the expander G_{exp} under edge updates in G, with constant-time slowdown.

Attempt: For any edge insertion uv in G, add two V-to-L edges. Maintain balanced degrees in L by choosing the minimum degree vertices in L.

This attempt fails because the minimum degree vertex might already be a neighbor of the affected vertex.



Solution: (greedy + lazy strategy):

- Greedy - choose the minimum degree non-neighbor in L. This might take more than constant time.
- Lazy - wait until enough updates are made before adding new neighbors to a vertex, to compenstate for costly updates. In addition, rebalance periodically.

Consequences

For the following problems: *k-Clique Detection, Maximum Matching, Dynamic Densest Subgraph, Max-Clique, Vertex Cover, Max-Cut, Minimum Dominating Set.*

- **Their worst-case instances are expanders even in the deterministic and dynamic settings.**
- **The expander decomposition method is not the key for solving these problems or derandomizing their best algorithms.**