Triangle Detection in H-Free Graphs

Amir Abboud, Ron Safier, *Nathan Wallheimer*

WEIZMANN INSTITUTE OF SCIENCE

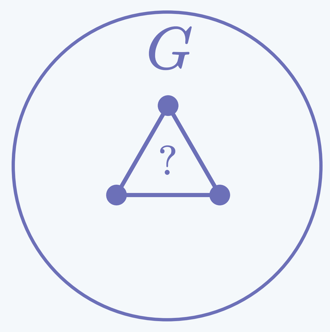
European Research Council
Established by the European Commission

Triangle Detection

Triangle Detection: Decide if an n-vertex graph G contains a triangle.

- **Trivial algorithm:** $O(n^3)$ time.
- **Fast Matrix Multiplication:** $O(n^{2.371\ldots})$ time.

No combinatorial $n^{3-\Omega(1)}$ algorithm is known!



Possible Implications of a Combinatorial $n^{3-\Omega(1)}$-Time Algorithm

A combinatorial algorithm has the potential to generalize to other variants of Triangle Detection that capture the hardness of some of the most fundamental problems.

- **Online Matrix-Vector Multiplication** $\xrightarrow{n^{3-\Omega(1)}}$ **Online Triangle Detection**
- **All Pairs Shortest Paths** $\xrightarrow{n^{3-\Omega(1)}}$ **Min-Weight Triangle Detection**
- **3-SUM** $\xrightarrow{n^{2-\Omega(1)}}$ **0-Weight Triangle Detection**

The Main Question

Is there a combinatorial $n^{3-\Omega(1)}$-time algorithm when G has more structure?

- **Our notion of structure:** H-free graphs.
- **Motivation:** An approach for combinatorial algorithms that exploits the dichotomy between pseudo-randomness and structure.

H-Free Graphs & Objective

Definition: A graph G is H-free if it does not contain H as a subgraph.

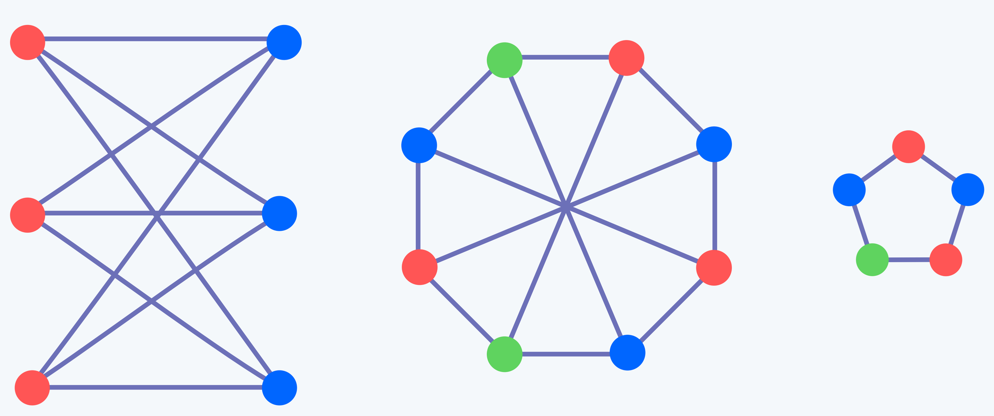
The Objective: Classify each pattern H into one of the following classes:

1. Triangle Detection in H-free graphs is **solvable in $n^{3-\Omega(1)}$ time** via a combinatorial algorithm.
2. Triangle Detection in H-free graphs is **as hard as in general graphs**.

Our Main Result: Easy Patterns

Theorem: There is an $O(n^{3-2^{-k}})$-time algorithm ($k = |V(H)|$) for patterns that admit a *degenerate* 3-*coloring*.

Definition: Pattern H is *degenerately colored* if every induced subgraph of H contains a vertex with a monochromatic neighborhood.



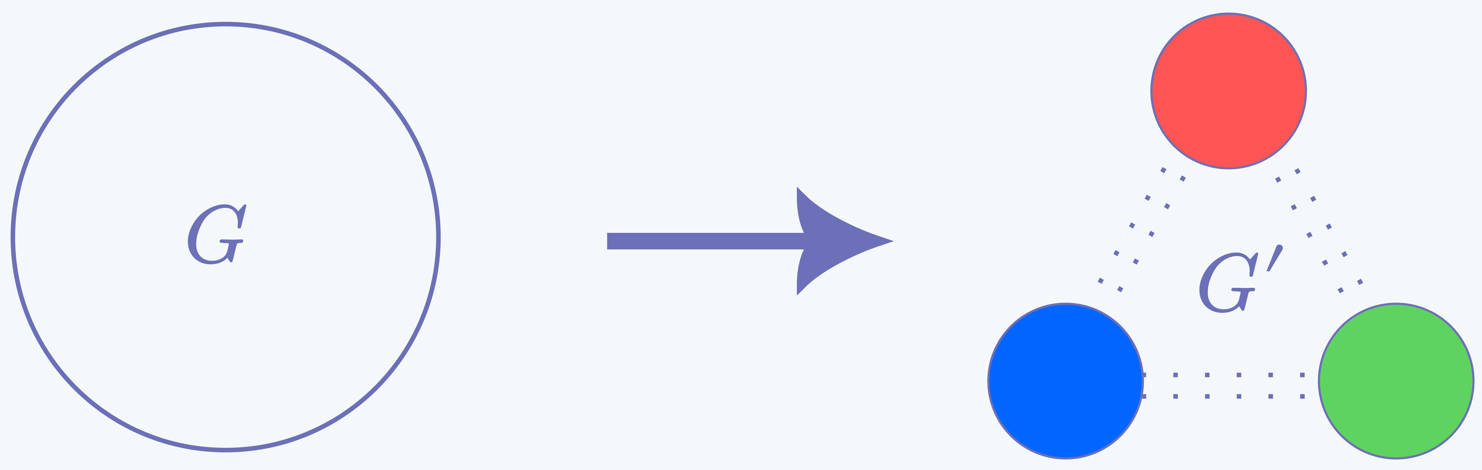
E.g., bipartite graphs, odd cycles, subgraphs of their blowups ...

High-level approach: Embed vertices of H with monochromatic neighborhoods one by one. Restrict G to neighborhoods of embedded vertices. The recursion terminates when reaching a locally-sparse subgraph of G that is trivial to solve in $n^{3-\Omega(1)}$ time.

The Hard Patterns

Triangle Detection in H-free graphs is **as hard as in general graphs** if:

- H is **not 3-colorable** (via Color-Coding), or
- H contains $\geq$ **2 triangles** (via Subsampling).



High-level approach: Partition the vertices into 3 buckets randomly. Delete edges within each bucket. Subsample each bucket with dyadic probabilities.

Consequences

- **Complete classification** of all patterns of size < 9.
- **Listing all triangles** in H-free graphs in the same running time.
- **Induced H-free graphs:** Algorithms for induced H-free graphs via a novel reduction from the induced setting to the subgraph one (follow-up work).

Scan the QR code for the follow-up paper about induced H-free graphs: