A Characterization of Easily Testable Induced Subgraphs (Part I)

Noga Alon and Asaf Shapira

Combinatorics, Probability and Computing **15** (2006) 791–805.

Speaker: Joseph, Chuang-Chieh Lin
Advisor: Professor Maw-Shang Chang

Computation Theory Laboratory
Dept. Computer Science and Information Engineering
National Chung Cheng University, Taiwan

December 3, 2008

What you need is that your brain is open.
– Paul Erdős

Outline

1. Introduction
2. Testing induced P_2-freeness
3. Testing induced P_3-freeness
4. Concluding remarks

Outline

1. Introduction
2. Testing induced P_2-freeness
3. Testing induced P_3-freeness
4. Concluding remarks

Introduction (model)

- Graph model: dense graph (adjacency matrix) for $G(V, E)$.
  - undirected, no self-loops, ≤ 1 edge between any $u, v \in V$
  - $|V| = n$ vertices and $|E| = \Omega(n^2)$ edges.
- A graph property:
  - A set of graphs closed under isomorphisms.
- Let $\mathbb{P}$ be a graph property.
  - ϵ-far from satisfying $\mathbb{P}$:
    - $\geq \epsilon n^2$ edges should be deleted or added to let the graph satisfy $\mathbb{P}$

Introduction (model)

- Graph model: dense graph (adjacency matrix) for $G(V, E)$.
  - undirected, no self-loops, ≤ 1 edge between any $u, v \in V$
  - $|V| = n$ vertices and $|E| = \Omega(n^2)$ edges.
- A graph property:
  - A set of graphs closed under isomorphisms.
- Let $\mathbb{P}$ be a graph property.
  - ϵ-far from satisfying $\mathbb{P}$:
    - $\geq \epsilon n^2$ edges should be deleted or added to let the graph satisfy $\mathbb{P}$

Introduction (model)

- Graph model: dense graph (adjacency matrix) for $G(V, E)$.
  - undirected, no self-loops, ≤ 1 edge between any $u, v \in V$
  - $|V| = n$ vertices and $|E| = \Omega(n^2)$ edges.
- A graph property:
  - A set of graphs closed under isomorphisms.
- Let $\mathbb{P}$ be a graph property.
  - ϵ-far from satisfying $\mathbb{P}$:
    - $\geq \epsilon n^2$ edges should be deleted or added to let the graph satisfy $\mathbb{P}$

Introduction (property testing)

- Property testing:
  - it does NOT precisely determine YES or NO for a decision problem;
  - requires sublinear running time
- A property tester for $\mathbb{P}$:
  - A randomized algorithm such that
    - it answers "YES" with probability of $\geq 2/3$ if G satisfies $\mathbb{P}$, and
    - it answers "NO" with probability of $\geq 2/3$ if G is ϵ-far from satisfying $\mathbb{P}$.

Introduction (property testing)

- Property testing:
  - it does NOT precisely determine YES or NO for a decision problem;
  - requires sublinear running time
- A property tester for $\mathbb{P}$:
  - A randomized algorithm such that
    - it answers "YES" with probability of $\geq 2/3$ if G satisfies $\mathbb{P}$, and
    - it answers "NO" with probability of $\geq 2/3$ if G is ϵ-far from satisfying $\mathbb{P}$.

Surveys...

- E. Fischer: The art of uninformed decisions: A primer to property testing. *The Computational Complexity Column of The Bulletin of the European Association for Theoretical Computer Science*, **75** (2001), pp. 97–126.
- O. Goldreich: *Combinatorial property testing - a survey*. Randomization Methods in Algorithm Design (P. M. Pardalos, S. Rajasekaran and J. D. P. Rolim eds.), AMS-DIMACS (1998), pp. 45–60.
- D. Ron: Property testing. *Handbook of Randomized Computing*, Vol. II, Kluwer Academic Publishers (P. M. Pardalos, S. Rajasekaran and J. D. P. Rolim eds.), 2001, pp. 597–649.

Introduction (testing graph properties)

- Throughout this talk, we focus on graph properties and the dense graph model.
- A property tester has the ability to make queries and then make decision by making use of the answers of queries.
  - To see whether a desired pair of vertices are adjacent or not.
- And, we care about query complexities in this talk.
- With a slight abuse of notation, $\log(n) = \ln(n)$.
- Assume that n is large enough and ϵ is small enough.

Introduction (testing graph properties)

- Throughout this talk, we focus on graph properties and the dense graph model.
- A property tester has the ability to make queries and then make decision by making use of the answers of queries.
  - To see whether a desired pair of vertices are adjacent or not.
- And, we care about query complexities in this talk.
- With a slight abuse of notation, $\log(n) = \ln(n)$.
- Assume that n is large enough and ϵ is small enough.

Introduction (testing graph properties)

- Throughout this talk, we focus on graph properties and the dense graph model.
- A property tester has the ability to make queries and then make decision by making use of the answers of queries.
  - To see whether a desired pair of vertices are adjacent or not.
- And, we care about query complexities in this talk.
- With a slight abuse of notation, $\log(n) = \ln(n)$.
- Assume that n is large enough and ϵ is small enough.

Introduction (testing graph properties)

- Throughout this talk, we focus on graph properties and the dense graph model.
- A property tester has the ability to make queries and then make decision by making use of the answers of queries.
  - To see whether a desired pair of vertices are adjacent or not.
- And, we care about query complexities in this talk.
- With a slight abuse of notation, $\log(n) = \ln(n)$.
- Assume that n is large enough and ϵ is small enough.

Introduction (testing graph properties)

- Throughout this talk, we focus on graph properties and the dense graph model.
- A property tester has the ability to make queries and then make decision by making use of the answers of queries.
  - To see whether a desired pair of vertices are adjacent or not.
- And, we care about query complexities in this talk.
- With a slight abuse of notation, $\log(n) = \ln(n)$.
- Assume that n is large enough and ϵ is small enough.

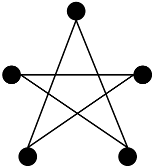
Introduction (testing graph properties)

- $\mathbb{P}$ is testable if
  - $\exists$ a property tester for $\mathbb{P}$ such that its query complexity is independent of n.
- $\mathbb{P}$ is called easily testable if
  - $\exists$ a property tester for $\mathbb{P}$ such that its query complexity is independent of n and **polynomial in** $1/\epsilon$.

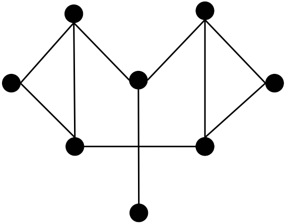
Introduction (testing graph properties)

- $\mathbb{P}$ is testable if
  - $\exists$ a property tester for $\mathbb{P}$ such that its query complexity is independent of n.
- $\mathbb{P}$ is called easily testable if
  - $\exists$ a property tester for $\mathbb{P}$ such that its query complexity is independent of n and **polynomial in** $1/\epsilon$.

Induced subgraph

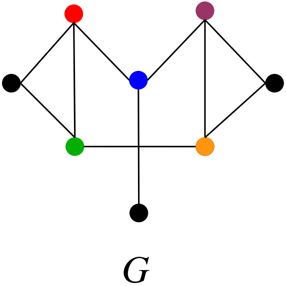


H

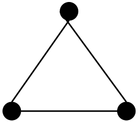


G

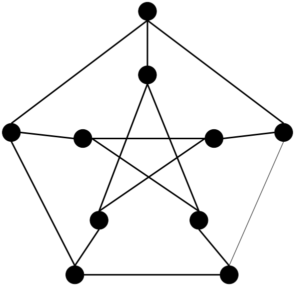
Induced subgraph



Induced subgraph (contd.)



H



G

Induced H-freeness

- $\mathbb{P}^*_H$: the property that a graph having no H as an induced subgraph.
- A graph G satisfies $\mathbb{P}^*_H \Leftrightarrow G$ does not have H as an induced subgraph.

Goals of this talk

- We show that $\mathbb{P}^*_{P_2}$ and $\mathbb{P}^*_{P_3}$ are easily testable.
- Testing $\mathbb{P}^*_{P_2}$ requires only $O(1/\epsilon)$ queries, and testing $\mathbb{P}^*_{P_3}$ requires $O(\log^2(1/\epsilon)/\epsilon^2)$ queries.
- The other important part of the paper will be introduced next time.

Goals of this talk

- We show that $\mathbb{P}^*_{P_2}$ and $\mathbb{P}^*_{P_3}$ are easily testable.
- Testing $\mathbb{P}^*_{P_2}$ requires only $O(1/\epsilon)$ queries, and testing $\mathbb{P}^*_{P_3}$ requires $O(\log^2(1/\epsilon)/\epsilon^2)$ queries.
- The other important part of the paper will be introduced next time.

Goals of this talk

- We show that $\mathbb{P}^*_{P_2}$ and $\mathbb{P}^*_{P_3}$ are easily testable.
- Testing $\mathbb{P}^*_{P_2}$ requires only $O(1/\epsilon)$ queries, and testing $\mathbb{P}^*_{P_3}$ requires $O(\log^2(1/\epsilon)/\epsilon^2)$ queries.
- The other important part of the paper will be introduced next time.

An easy example: $H = \mathbb{P}^*_{P_2}$

- Testing $\mathbb{P}^*_{P_2}$ $\Leftrightarrow$ testing emptiness of a graph.
  - Query complexity and time complexity: $O(1/\epsilon)$
  - How can it be done?

An easy example: $H = \mathbb{P}^*_{P_2}$

- Testing $\mathbb{P}^*_{P_2}$ $\Leftrightarrow$ testing emptiness of a graph.
  - Query complexity and time complexity: $O(1/\epsilon)$
  - How can it be done?

An easy example: $H = \mathbb{P}^*_{P_2}$

- Testing $\mathbb{P}^*_{P_2}$ $\Leftrightarrow$ testing emptiness of a graph.
  - Query complexity and time complexity: $O(1/\epsilon)$
  - How can it be done?

An easy example: $H = \mathbb{P}^*_{P_2}$ (contd.)

- ϵ-far from being $\mathbb{P}_{P_2}$:
  - ϵn^2 pairs of vertices are *adjacent*.
- A property tester works as follows.
  - Repeatedly, for $2/\epsilon$ times, pick two vertices uniformly at random and check if they are adjacent. Once an edge is found, return NO, otherwise (i.e., all of the chosen pairs of vertices are not adjacent) return YES.

An easy example: $H = \mathbb{P}^*_{P_2}$ (contd.)

- ϵ-far from being $\mathbb{P}_{P_2}$:
  - ϵn^2 pairs of vertices are *adjacent*.
- A property tester works as follows.
  - Repeatedly, for $2/\epsilon$ times, pick two vertices uniformly at random and check if they are adjacent. Once an edge is found, return NO, otherwise (i.e., all of the chosen pairs of vertices are not adjacent) return YES.

An easy example: $H = \mathbb{P}^*_{P_2}$ (contd.)

- ϵ-far from being $\mathbb{P}_{P_2}$:
  - ϵn^2 pairs of vertices are *adjacent*.
- A property tester works as follows.
  - Repeatedly, for $2/\epsilon$ times, pick two vertices uniformly at random and check if they are adjacent. Once an edge is found, return NO, otherwise (i.e., all of the chosen pairs of vertices are not adjacent) return YES.

An easy example: $H = \mathbb{P}^*_{P_2}$ (contd.)

- **Pr**[the property tester returns YES |G satisfies $\mathbb{P}^*_{P_2}$] = 1.
- **Pr**[the property tester returns YES |G is ϵ-far from satisfying $\mathbb{P}^*_{P_2}$] = $(1 - \epsilon n^2/n^2)^{2/\epsilon} = (1-\epsilon)^{2/\epsilon} < e^{-2} < 1/3$.

⋆ Note that $\lim_{\epsilon \longrightarrow 0}(1-\epsilon)^{1/\epsilon} = e^{-1}$.

An easy example: $H = \mathbb{P}^*_{P_2}$ (contd.)

- **Pr**[the property tester returns YES |G satisfies $\mathbb{P}^*_{P_2}$] $= 1$.
- **Pr**[the property tester returns YES |G is ϵ-far from satisfying $\mathbb{P}^*_{P_2}$] $= (1 - \epsilon n^2/n^2)^{2/\epsilon} = (1-\epsilon)^{2/\epsilon} < e^{-2} < 1/3$.

⋆ Note that $\lim_{\epsilon \to 0}(1-\epsilon)^{1/\epsilon} = e^{-1}$.

Outline

1. Introduction
2. Testing induced P_2-freeness
3. Testing induced P_3-freeness
4. Concluding remarks

The property tester for $\mathbb{P}^*_{P_3}$

- The property tester is as follows:
  1. Pick a random subset of $10\log(1/\epsilon)/\epsilon$ vertices.
  2. Check if there is an induced copy of P_3 spanned by this set.
- The query complexity is at most $O(\log^2(1/\epsilon)/\epsilon^2)$.
- If G satisfies $\mathbb{P}^*_{P_3}$, the algorithm always answers correctly (i.e., answers YES since there is no induced P_3).
- We have to show that if G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, the algorithm finds an induced copy of P_3 with probability $\geq 2/3$.

The property tester for $\mathbb{P}^*_{P_3}$

- The property tester is as follows:
  1. Pick a random subset of $10\log(1/\epsilon)/\epsilon$ vertices.
  2. Check if there is an induced copy of P_3 spanned by this set.
- The query complexity is at most $O(\log^2(1/\epsilon)/\epsilon^2)$.
- If G satisfies $\mathbb{P}^*_{P_3}$, the algorithm always answers correctly (i.e., answers YES since there is no induced P_3).
- We have to show that if G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, the algorithm finds an induced copy of P_3 with probability $\geq 2/3$.

The property tester for $\mathbb{P}^*_{P_3}$

- The property tester is as follows:
  1. Pick a random subset of $10\log(1/\epsilon)/\epsilon$ vertices.
  2. Check if there is an induced copy of P_3 spanned by this set.
- The query complexity is at most $O(\log^2(1/\epsilon)/\epsilon^2)$.
- If G satisfies $\mathbb{P}^*_{P_3}$, the algorithm always answers correctly (i.e., answers YES since there is no induced P_3).
- We have to show that if G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, the algorithm finds an induced copy of P_3 with probability $\geq 2/3$.

The property tester for $\mathbb{P}^*_{P_3}$

- The property tester is as follows:
  1. Pick a random subset of $10\log(1/\epsilon)/\epsilon$ vertices.
  2. Check if there is an induced copy of P_3 spanned by this set.
- The query complexity is at most $O(\log^2(1/\epsilon)/\epsilon^2)$.
- If G satisfies $\mathbb{P}^*_{P_3}$, the algorithm always answers correctly (i.e., answers YES since there is no induced P_3).
- We have to show that if G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, the algorithm finds an induced copy of P_3 with probability $\geq 2/3$.

The property tester for $\mathbb{P}^*_{P_3}$

- The property tester is as follows:
  1. Pick a random subset of $10 \log(1/\epsilon)/\epsilon$ vertices.
  2. Check if there is an induced copy of P_3 spanned by this set.
- The query complexity is at most $O(\log^2(1/\epsilon)/\epsilon^2)$.
- If G satisfies $\mathbb{P}^*_{P_3}$, the algorithm always answers correctly (i.e., answers YES since there is no induced P_3).
- We have to show that if G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, the algorithm finds an induced copy of P_3 with probability $\geq 2/3$.

High degree vertices

- Let *HIGH* be the set $\{v \in V(G) \mid \deg(v) \geq \frac{\epsilon n}{4}\}$.
  - Intuitively, vertices of HIGH have high contribution to G being ϵ-far from satisfying $\mathbb{P}^*_{P_3}$.

HIGH has high contribution indeed!

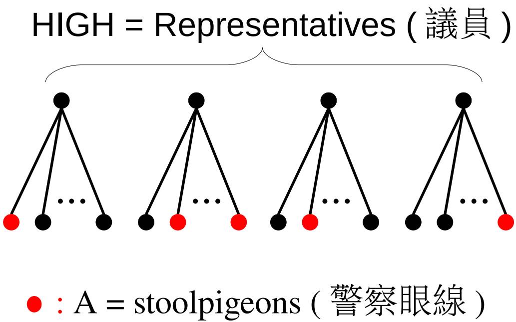
Claim 1

Assume that G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$. Let $W \subseteq V(G)$ contain at least $|HIGH| - \frac{\epsilon}{4}n$ vertices of HIGH, then the induced subgraph of G on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$.

Randomly chosen subset of vertices are Good w.h.p.

Definition 3.1

We call a set $A \subseteq V(G)$ Good if at least $|\text{HIGH}| - \frac{\epsilon}{4}n$ vertices of HIGH have a neighbor in A.



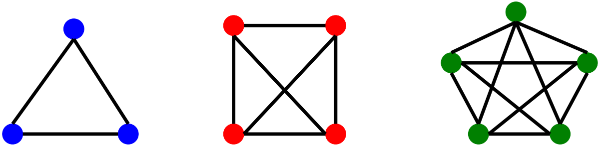
Randomly chosen subset of vertices are Good w.h.p.

Claim 2

A randomly chosen subset $A \subseteq V(G)$ of size $8\log(1/\epsilon)/\epsilon$ is Good with probability at least $7/8$.

A well-known observation for induced P_3-free graphs

- A graph is induced P_3-free if and only if it is disjoint union of cliques.



Continue to show that the property tester for $\mathbb{P}^*_{P_3}$ is valid

- First we choose a random subset $A \subset V$ of size $8\log(1/\epsilon)/\epsilon$.
  - That is, equivalently, eying on part of the vertices randomly chosen by the algorithm.
- Assume that A is Good (this is not true with probability at most $1/8$).
- If A contains an induced copy of P_3, then we are done.

Continue to show that the property tester for $\mathbb{P}^*_{P_3}$ is valid

- First we choose a random subset $A \subset V$ of size $8 \log(1/\epsilon)/\epsilon$.
  - That is, equivalently, eying on part of the vertices randomly chosen by the algorithm.
- Assume that A is Good (this is not true with probability at most $1/8$).
- If A contains an induced copy of P_3, then we are done.

Continue to show that the property tester for $\mathbb{P}^*_{P_3}$ is valid

- First we choose a random subset $A \subset V$ of size $8\log(1/\epsilon)/\epsilon$.
  - That is, equivalently, eying on part of the vertices randomly chosen by the algorithm.
- Assume that A is Good (this is not true with probability at most $1/8$).
- If A contains an induced copy of P_3, then we are done.

Continue to show that the property tester for $\mathbb{P}^*_{P_3}$ is valid

- First we choose a random subset $A \subset V$ of size $8\log(1/\epsilon)/\epsilon$.
  - That is, equivalently, eying on part of the vertices randomly chosen by the algorithm.
- Assume that A is Good (this is not true with probability at most $1/8$).
- If A contains an induced copy of P_3, then we are done.

Keep proving...

- Otherwise, (i.e., A contains no induced copy of P_3)
  - Let W be the set of all the vertices $v \in V$ that ≥ 1 neighbor in A.
  - The induced subgraph on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$. (WHY?)
  - Recall that G is assume to be ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, and A is Good.
- And of course, we can assume that A can be partitioned into disjoint union of cliques $C_1, C_2, \ldots, C_r$, for some integer r.

Keep proving...

- Otherwise, (i.e., A contains no induced copy of P_3)
  - Let W be the set of all the vertices $v \in V$ that ≥ 1 neighbor in A.
  - The induced subgraph on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$. (WHY?)
  - Recall that G is assume to be ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, and A is Good.
- And of course, we can assume that A can be partitioned into disjoint union of cliques $C_1, C_2, \ldots, C_r$, for some integer r.

Keep proving...

- Otherwise, (i.e., A contains no induced copy of P_3)
  - Let W be the set of all the vertices $v \in V$ that ≥ 1 neighbor in A.
  - The induced subgraph on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$. (WHY?)
  - Recall that G is assume to be ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, and A is Good.
- And of course, we can assume that A can be partitioned into disjoint union of cliques $C_1, C_2, \ldots, C_r$, for some integer r.

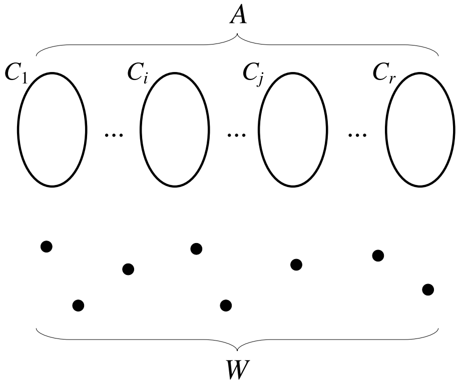
Keep proving...

- Otherwise, (i.e., A contains no induced copy of P_3)
  - Let W be the set of all the vertices $v \in V$ that ≥ 1 neighbor in A.
  - The induced subgraph on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$. (WHY?)
  - Recall that G is assume to be ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, and A is Good.
- And of course, we can assume that A can be partitioned into disjoint union of cliques $C_1, C_2, \ldots, C_r$, for some integer r.

Keep proving...

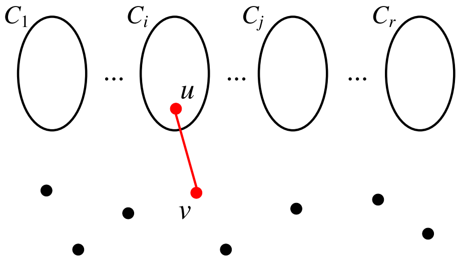
- Otherwise, (i.e., A contains no induced copy of P_3)
  - Let W be the set of all the vertices $v \in V$ that ≥ 1 neighbor in A.
  - The induced subgraph on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$. (WHY?)
  - Recall that G is assume to be ϵ-far from satisfying $\mathbb{P}^*_{P_3}$, and A is Good.
- And of course, we can assume that A can be partitioned into disjoint union of cliques $C_1, C_2, \ldots, C_r$, for some integer r.

Keep proving...



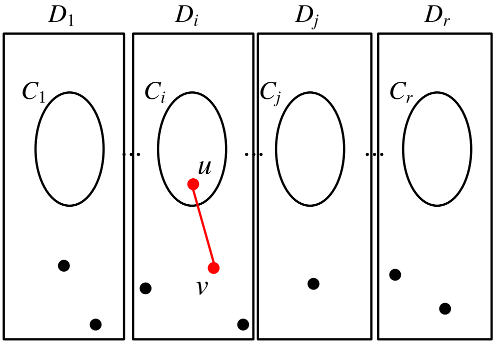
Keep proving...

- If a vertex $v \in W$ is connected to $u \in C_i \subseteq A$, it follows that if W can be partitioned into cliques $D_1, \ldots, D_r$, where for $1 \leq i \leq r$, $C_i \subseteq D_i$, then v would have to belong to D_i.



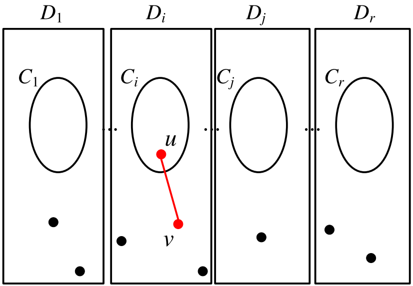
Keep proving...

- If a vertex $v \in W$ is connected to $u \in C_i \subseteq A$, it follows that if W can be partitioned into cliques $D_1, \ldots, D_r$, where for $1 \leq i \leq r$, $C_i \subseteq D_i$, then v would have to belong to D_i.



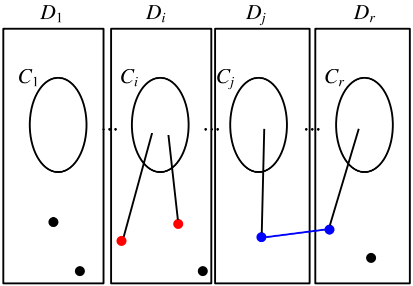
Keep proving...

- For each $v \in W$ connected to $u \in C_i$, assign v the number i. If v is connected to vertices that belong to different C_i's, then assign v any of these numbers.
- The numbering induces a partition of W into r subsets.



Keep proving...

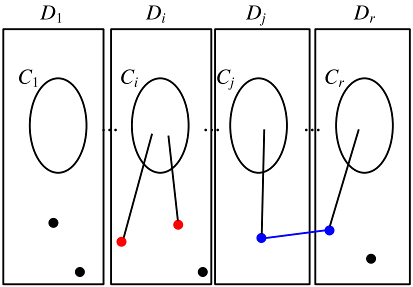
- Violating pairs: "$s, t \in D_i$ but s, t are not connected" or "$s \in D_i, t \in D_j$ for $i \neq j$ but s, t are connected".
- As W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$, there are at least $\frac{\epsilon}{2}n^2$ violating pairs of vertices in W.



Keep proving...

- Therefore, choosing a set B of $8/\epsilon$ randomly chosen pairs of vertices fails to find violating pairs with probability of at most

$$\left(1 - \frac{\epsilon n^2/2}{n(n-1)/2}\right)^{8/\epsilon} < \left(1 - \frac{\epsilon}{2}\right)^{8/\epsilon} < e^{-4} < \frac{1}{8}.$$



To sum up

- By Claim 2, **Pr**[A is NOT Good] $\leq \frac{1}{8}$.
- **Pr**[B does NOT contain any violating pair of vertices] $\leq \frac{1}{8}$.
- Hence with probability at least $1 - \frac{1}{8} - \frac{1}{8} = \frac{3}{4}$ the induced subgraph on $A \cup B$ is not induced P_3-free.
- Since $|A| + |B| = O(8 \log(1/\epsilon)/\epsilon + 8/\epsilon) = O(8 \log(1/\epsilon)/\epsilon)$, the proof is complete!

To sum up

- By Claim 2, **Pr**[A is NOT Good] $\leq \frac{1}{8}$.
- **Pr**[B does NOT contain any violating pair of vertices] $\leq \frac{1}{8}$.
- Hence with probability at least $1 - \frac{1}{8} - \frac{1}{8} = \frac{3}{4}$ the induced subgraph on $A \cup B$ is not induced P_3-free.
- Since $|A| + |B| = O(8\log(1/\epsilon)/\epsilon + 8/\epsilon) = O(8\log(1/\epsilon)/\epsilon)$, the proof is complete!

To sum up

- By Claim 2, $\mathbf{Pr}[A \text{ is NOT Good}] \leq \frac{1}{8}$.
- $\mathbf{Pr}[B \text{ does NOT contain any violating pair of vertices}] \leq \frac{1}{8}$.
- Hence with probability at least $1 - \frac{1}{8} - \frac{1}{8} = \frac{3}{4}$ the induced subgraph on $A \cup B$ is not induced P_3-free.
- Since $|A| + |B| = O(8 \log(1/\epsilon)/\epsilon + 8/\epsilon) = O(8 \log(1/\epsilon)/\epsilon)$, the proof is complete!

To sum up

- By Claim 2, $\mathbf{Pr}[A \text{ is NOT Good}] \leq \frac{1}{8}$.
- $\mathbf{Pr}[B \text{ does NOT contain any violating pair of vertices}] \leq \frac{1}{8}$.
- Hence with probability at least $1 - \frac{1}{8} - \frac{1}{8} = \frac{3}{4}$ the induced subgraph on $A \cup B$ is not induced P_3-free.
- Since $|A| + |B| = O(8 \log(1/\epsilon)/\epsilon + 8/\epsilon) = O(8 \log(1/\epsilon)/\epsilon)$, the proof is complete!

The search for truth is more precious than its possession.
– Albert Einstein

Painful time starts now

Let us go back to the proofs of the previous claims.

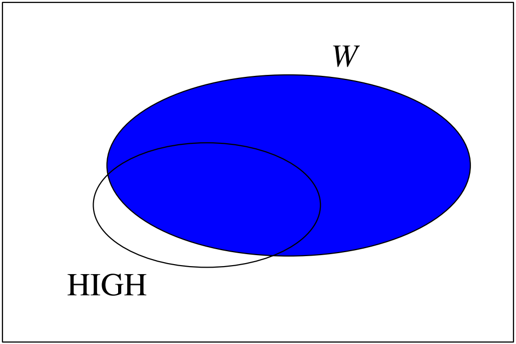
Proof of Claim 1

Claim 1

Assume that G is ϵ-far from satisfying $\mathbb{P}^*_{P_3}$. Let $W \subseteq V(G)$ contain at least $|\mathrm{HIGH}| - \frac{\epsilon}{4}n$ vertices of HIGH, then the induced subgraph of G on W is at least $\frac{\epsilon}{2}$-far from satisfying $\mathbb{P}^*_{P_3}$.

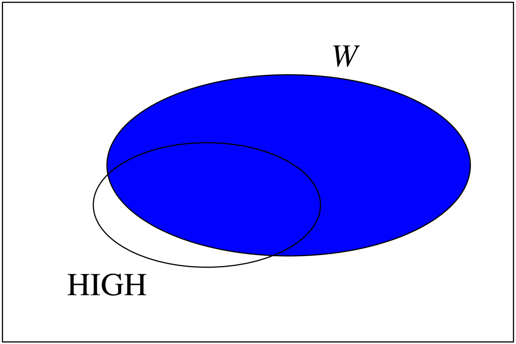
Proof of Claim 1 (contd.)

- Assume this is not the case (proof by contradiction).



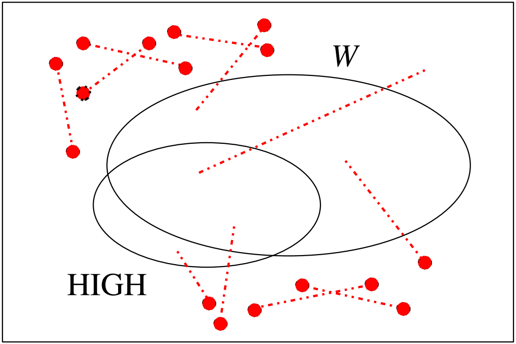
Proof of Claim 1 (contd.)

- That is, we can make less than $\frac{\epsilon}{2}n^2$ changes (edge deletions or edge additions) within W and get a graph that contains no induced copy of P_3 within W.



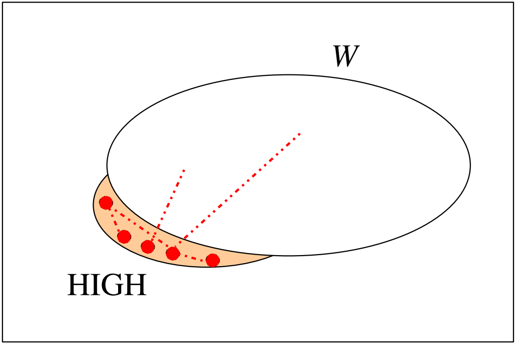
Proof of Claim 1 (contd.)

- Then we remove all the edges touching a vertex not in $W \cup \mathrm{HIGH}$.
- $\leq n \cdot \frac{\epsilon}{4} n$ such edges.



Proof of Claim 1 (contd.)

- Then we remove any edge touching a vertex in HIGH $\setminus$ W.
- $\leq \frac{\epsilon}{4}n \cdot n$ such edges since $|\text{HIGH} \setminus W| \leq \frac{\epsilon}{4}n$.



Proof of Claim 1 (contd.)

- Thus we obtain a graph that satisfies $\mathbb{P}^*_{P_3}$.
- $< \epsilon n^2$ edges are added or deleted in G, so G is not ϵ-far from satisfying $\mathbb{P}^*_{P_3}$.
  - This contradicts the assumption!

Proof of Claim 1 (contd.)

- Thus we obtain a graph that satisfies $\mathbb{P}^*_{P_3}$.
- $< \epsilon n^2$ edges are added or deleted in G, so G is not ϵ-far from satisfying $\mathbb{P}^*_{P_3}$.
    - This contradicts the assumption!

Proof of Claim 2

Claim 2

A randomly chosen subset $A \subseteq V(G)$ of size $8\log(1/\epsilon)/\epsilon$ is Good with probability at least $7/8$.

Proof of Claim 2 (contd.)

- Let A be a randomly chosen subset of size $8\log(1/\epsilon)/\epsilon$.
- Consider a vertex $v \in$ HIGH.
- Since v has at least $\frac{\epsilon}{4}n$ neighbors, the probability that A does not contain any neighbor of v is at most

$$\left(1-\frac{\epsilon}{4}\right)^{8\log(1/\epsilon)/\epsilon} = \left[\left(1-\frac{\epsilon}{4}\right)^{\frac{-4}{\epsilon}}\right]^{-2\log(\frac{1}{\epsilon})} \leq e^{\log\epsilon^2} = \epsilon^2 \leq \frac{\epsilon}{32},$$

where we assume that $\epsilon < 1/32$.

▷ Exercise: Show that the above assumption can be loosed to $\epsilon < 1$ by letting $|A| = \frac{4\log(1/\epsilon)}{\epsilon} + \frac{20}{\epsilon}$.

Proof of Claim 2 (contd.)

- Let A be a randomly chosen subset of size $8\log(1/\epsilon)/\epsilon$.
- Consider a vertex $v \in$ HIGH.
- Since v has at least $\frac{\epsilon}{4}n$ neighbors, the probability that A does not contain any neighbor of v is at most

$$\left(1-\frac{\epsilon}{4}\right)^{8\log(1/\epsilon)/\epsilon} = \left[\left(1-\frac{\epsilon}{4}\right)^{\frac{-4}{\epsilon}}\right]^{-2\log(\frac{1}{\epsilon})} \leq e^{\log \epsilon^2} = \epsilon^2 \leq \frac{\epsilon}{32},$$

where we assume that $\epsilon < 1/32$.

▷ Exercise: Show that the above assumption can be loosed to $\epsilon < 1$ by letting $|A| = \frac{4\log(1/\epsilon)}{\epsilon} + \frac{20}{\epsilon}$.

Proof of Claim 2 (contd.)

- Let A be a randomly chosen subset of size $8\log(1/\epsilon)/\epsilon$.
- Consider a vertex $v \in$ HIGH.
- Since v has at least $\frac{\epsilon}{4}n$ neighbors, the probability that A does not contain any neighbor of v is at most

$$\left(1-\frac{\epsilon}{4}\right)^{8\log(1/\epsilon)/\epsilon} = \left[\left(1-\frac{\epsilon}{4}\right)^{\frac{-4}{\epsilon}}\right]^{-2\log(\frac{1}{\epsilon})} \leq e^{\log \epsilon^2} = \epsilon^2 \leq \frac{\epsilon}{32},$$

where we assume that $\epsilon < 1/32$.

▷ Exercise: Show that the above assumption can be loosed to $\epsilon < 1$ by letting $|A| = \frac{4\log(1/\epsilon)}{\epsilon} + \frac{20}{\epsilon}$.

Proof of Claim 2 (contd.)

- Let A be a randomly chosen subset of size $8\log(1/\epsilon)/\epsilon$.
- Consider a vertex $v \in \text{HIGH}$.
- Since v has at least $\frac{\epsilon}{4}n$ neighbors, the probability that A does not contain any neighbor of v is at most

$$\left(1-\frac{\epsilon}{4}\right)^{8\log(1/\epsilon)/\epsilon} = \left[\left(1-\frac{\epsilon}{4}\right)^{\frac{-4}{\epsilon}}\right]^{-2\log(\frac{1}{\epsilon})} \leq e^{\log\epsilon^2} = \epsilon^2 \leq \frac{\epsilon}{32},$$

where we assume that $\epsilon < 1/32$.

▷ Exercise: Show that the above assumption can be loosed to $\epsilon < 1$ by letting $|A| = \frac{4\log(1/\epsilon)}{\epsilon} + \frac{20}{\epsilon}$.

Proof of Claim 2 (contd.)

- We just obtained for $v \in$ HIGH,
  $\mathbf{Pr}[A$ does not contain any neighbor of $v] \leq \frac{\epsilon}{32}$.
- Let X denote the number of vertices that belong to HIGH and have no neighbor in A.
- Since $|\text{HIGH}| \leq n$, we have $\mathbf{E}[X] \leq \frac{\epsilon}{32} \cdot n$ (by linearity of expectation).
- By Markov's inequality, $\mathbf{Pr}[X \geq \frac{\epsilon}{4} n] \leq \frac{\mathbf{E}[X]}{\frac{\epsilon}{4} n} \leq \frac{\epsilon n/32}{\epsilon n/4} = 1/8$.
- Hence the proof is done.

Proof of Claim 2 (contd.)

- We just obtained for $v \in$ HIGH,
  $\mathbf{Pr}[A$ does not contain any neighbor of $v] \leq \frac{\epsilon}{32}$.
- Let X denote the number of vertices that belong to HIGH and have no neighbor in A.
- Since $|\text{HIGH}| \leq n$, we have $\mathbf{E}[X] \leq \frac{\epsilon}{32} \cdot n$ (by linearity of expectation).
- By Markov's inequality, $\mathbf{Pr}[X \geq \frac{\epsilon}{4} n] \leq \frac{\mathbf{E}[X]}{\frac{\epsilon}{4} n} \leq \frac{\epsilon n/32}{\epsilon n/4} = 1/8$.
- Hence the proof is done.

Proof of Claim 2 (contd.)

- We just obtained for $v \in$ HIGH,
  $\mathbf{Pr}[A$ does not contain any neighbor of $v] \leq \frac{\epsilon}{32}$.
- Let X denote the number of vertices that belong to HIGH and have no neighbor in A.
- Since $|\text{HIGH}| \leq n$, we have $\mathbf{E}[X] \leq \frac{\epsilon}{32} \cdot n$ (by linearity of expectation).
- By Markov's inequality, $\mathbf{Pr}[X \geq \frac{\epsilon}{4} n] \leq \frac{\mathbf{E}[X]}{\frac{\epsilon}{4} n} \leq \frac{\epsilon n/32}{\epsilon n/4} = 1/8$.
- Hence the proof is done.

Proof of Claim 2 (contd.)

- We just obtained for $v \in \text{HIGH}$,
  $\mathbf{Pr}[A \text{ does not contain any neighbor of } v] \leq \frac{\epsilon}{32}$.
- Let X denote the number of vertices that belong to HIGH and have no neighbor in A.
- Since $|\text{HIGH}| \leq n$, we have $\mathbf{E}[X] \leq \frac{\epsilon}{32} \cdot n$ (by linearity of expectation).
- By Markov's inequality, $\mathbf{Pr}[X \geq \frac{\epsilon}{4} n] \leq \frac{\mathbf{E}[X]}{\frac{\epsilon}{4} n} \leq \frac{\epsilon n/32}{\epsilon n/4} = 1/8$.
- Hence the proof is done.

Open problems

- Are P_4 and C_4 easily testable?

Thank you!