Testing Graph Properties with the Container Method

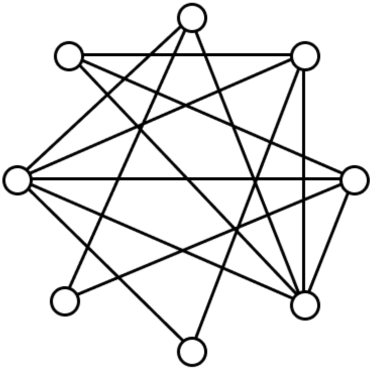
Eric Blais   Cameron Seth

June 5 2024

UNIVERSITY OF WATERLOO

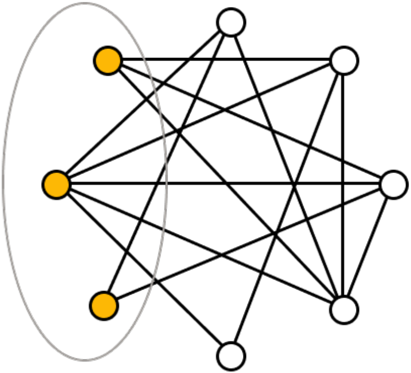
Independent Set Problem

Does a graph on n vertices have an independent set of size ρn?



Independent Set Problem

Does a graph on n vertices have an independent set of size ρn?

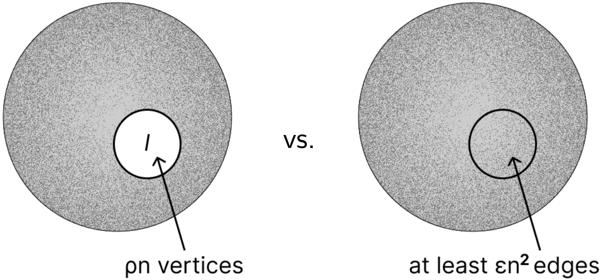


Testing Independent Sets

Testing Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn in G has at least ϵn^2 edges.

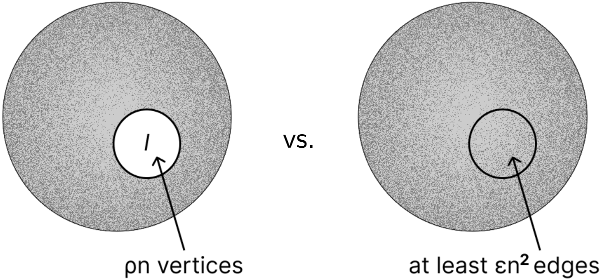


Testing Independent Sets

Testing Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn in G has at least ϵn^2 edges.



Theorem: Inspecting a random subgraph on $\widetilde{O}(\rho/\epsilon^4)$ vertices is sufficient to distinguish between (i) and (ii).

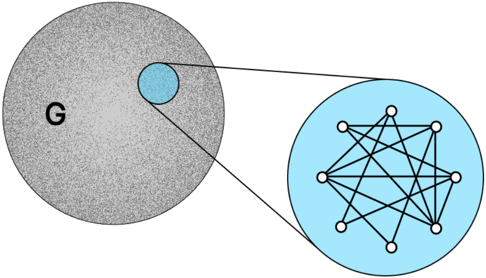
[Goldreich, Goldwasser, Ron '98]

Definitions

An ϵ-tester for ρ-IndepSet is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with high probability):

(i) G has a ρn independent set, and

(ii) every subgraph of size ρn in G has at least ϵn^2 edges (ϵ-far).

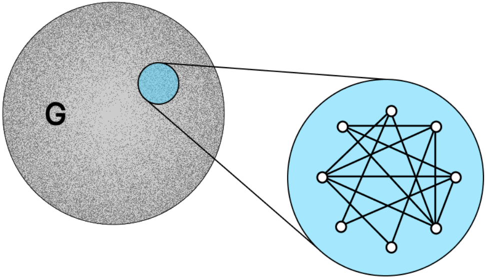


Definitions

An ϵ-tester for ρ-IndepSet is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with high probability):

(i) G has a ρn independent set, and

(ii) every subgraph of size ρn in G has at least ϵn^2 edges (ϵ-far).



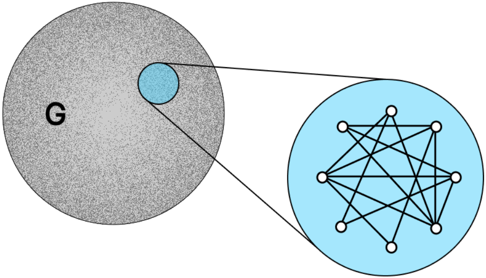
s is the sample complexity of the tester.

Definitions

An ϵ-tester for ρ-IndepSet is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with high probability):

(i) G has a ρn independent set, and

(ii) every subgraph of size ρn in G has at least ϵn^2 edges (ϵ-far).



s is the sample complexity of the tester.

Question: What is the minimum sample complexity $\mathcal{S}_{\rho\text{-IndepSet}}(n, \epsilon)$ of ϵ-testing the ρ-IndepSet property?

Results

Prior Results: The sample complexity $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon)$ for ϵ-testing ρ-$\textsc{IndepSet}$ is bounded above by:

- $\widetilde{O}(\rho/\epsilon^4)$ [Goldreich, Goldwasser, Ron '98]
- $\widetilde{O}(\rho^4/\epsilon^3)$ [Feige, Langberg, Schechtman '04]

and it is bounded below by:

- $\widetilde{\Omega}(\rho^3/\epsilon^2)$. [Feige, Langberg, Schechtman '04]

Results

Prior Results: The sample complexity $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon)$ for ϵ-testing ρ-$\textsc{IndepSet}$ is bounded above by:

- $\widetilde{O}(\rho/\epsilon^4)$ [Goldreich, Goldwasser, Ron '98]
- $\widetilde{O}(\rho^4/\epsilon^3)$ [Feige, Langberg, Schechtman '04]

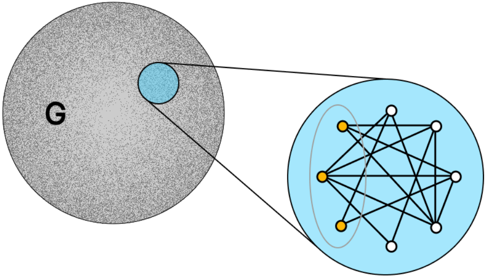
and it is bounded below by:

- $\widetilde{\Omega}(\rho^3/\epsilon^2)$. [Feige, Langberg, Schechtman '04]

Our Result: $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{\Theta}(\rho^3/\epsilon^2)$.

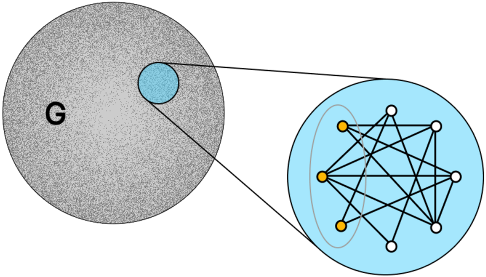
Proof Overview

Testing Algorithm: Take a random sample S of s vertices, check if the induced subgraph $G[S]$ has a ρs independent set.



Proof Overview

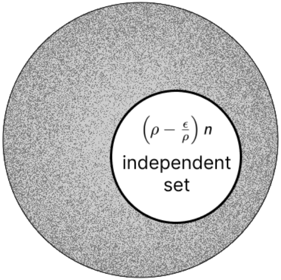
Testing Algorithm: Take a random sample S of s vertices, check if the induced subgraph $G[S]$ has a ρs independent set.



Key Challenge: Show that if G is ϵ-far from having a ρn independent set then a random induced subgraph on $s \approx \rho^3/\epsilon^2$ vertices has a ρs independent set with only small probability.

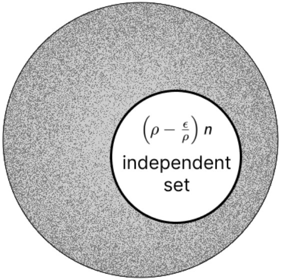
A Naive Approach

Observation: If G is ϵ-far from having a ρn independent set then the largest independent set in G is of size roughly $\left(\rho - \frac{\epsilon}{\rho}\right) n$.



A Naive Approach

Observation: If G is ϵ-far from having a ρn independent set then the largest independent set in G is of size roughly $\left(\rho - \frac{\epsilon}{\rho}\right) n$.

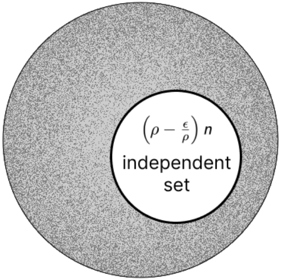


Approach:

- Upper bound the probability of getting ρs vertices in a sample of size s from a single $\left(\rho - \frac{\epsilon}{\rho}\right) n$ independent set.

A Naive Approach

Observation: If G is ϵ-far from having a ρn independent set then the largest independent set in G is of size roughly $\left(\rho - \frac{\epsilon}{\rho}\right) n$.

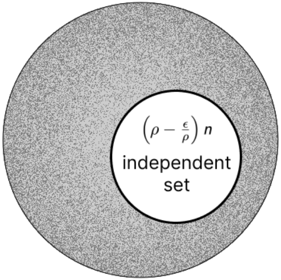


Approach:

- Upper bound the probability of getting ρs vertices in a sample of size s from a single $\left(\rho - \frac{\epsilon}{\rho}\right) n$ independent set.
- Union bound over all possible $\binom{n}{\left(\rho - \frac{\epsilon}{\rho}\right) n}$ independent sets.

A Naive Approach

Observation: If G is ϵ-far from having a ρn independent set then the largest independent set in G is of size roughly $\left(\rho - \frac{\epsilon}{\rho}\right) n$.

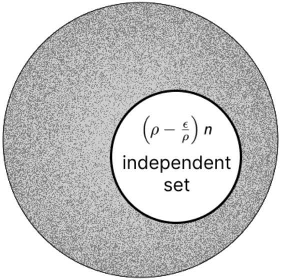


Approach:

- Upper bound the probability of getting ρs vertices in a sample of size s from a single $\left(\rho - \frac{\epsilon}{\rho}\right) n$ independent set.
- Union bound over all possible $\binom{n}{\left(\rho - \frac{\epsilon}{\rho}\right) n}$ independent sets.

A Naive Approach

Observation: If G is ϵ-far from having a ρn independent set then the largest independent set in G is of size roughly $\left(\rho - \frac{\epsilon}{\rho}\right) n$.



Approach:

- Upper bound the probability of getting ρs vertices in a sample of size s from a single $\left(\rho - \frac{\epsilon}{\rho}\right) n$ independent set.
- Union bound over all possible $\binom{n}{\left(\rho - \frac{\epsilon}{\rho}\right) n}$ independent sets.

Solution: The Graph Container Method

Graph Container Method

An Initial Graph Container Lemma

An Initial Graph Container Lemma

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-$\textsc{IndepSet}$. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

[Kleitman, Winston '82]

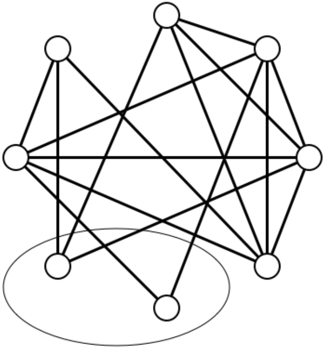
An Initial Graph Container Lemma

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-IndepSet. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

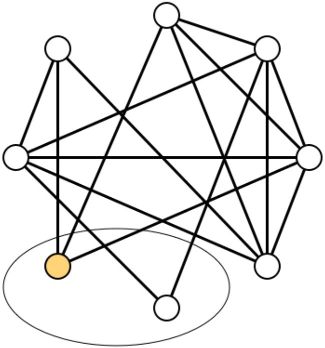
[Kleitman, Winston '82]

Graph Container Method – Encoding View



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

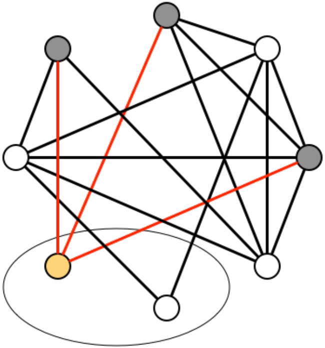
Graph Container Method – Encoding View



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

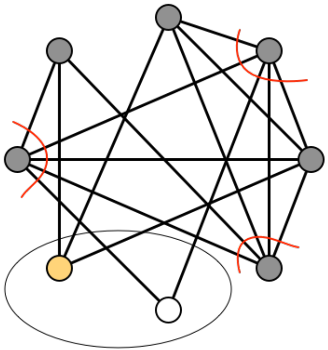
Graph Container Method – Encoding View



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Graph Container Method – Encoding View



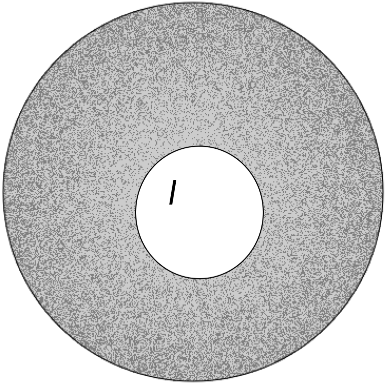
How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Container Generation Algorithm

Input: Graph G and an independent set I

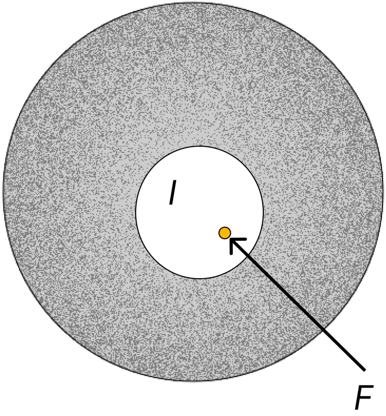
- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.



Container Generation Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

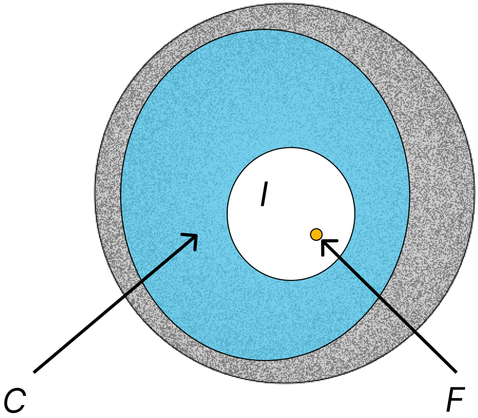


1st Iteration

Container Generation Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

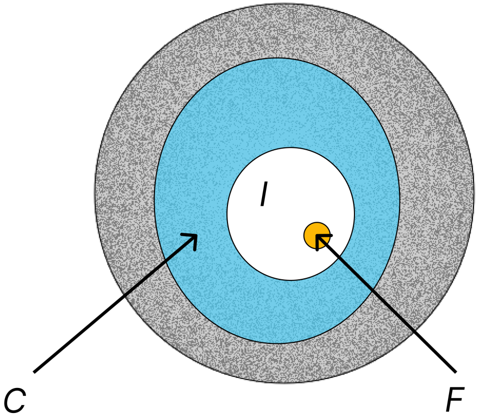


1st Iteration

Container Generation Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

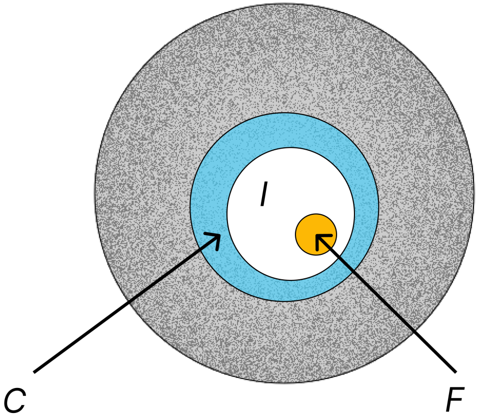


2nd Iteration

Container Generation Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

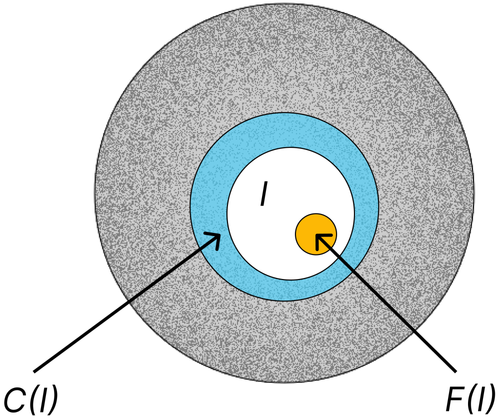


Final Iteration

Container Generation Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

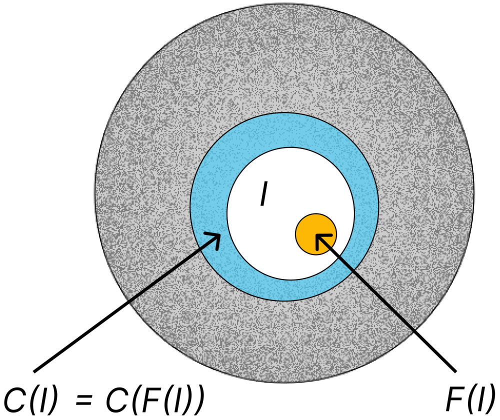


Final Iteration

Container Generation Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.



Final Iteration

Graph Container Lemma - Restated

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-$\textsc{IndepSet}$. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

[Kleitman, Winston '82]

Graph Container Lemma - Restated

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-$\textsc{IndepSet}$. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

[Kleitman, Winston '82]

Question: How do we construct the *collection* of containers?

Graph Container Lemma - Restated

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-$\textsc{IndepSet}$. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

[Kleitman, Winston '82]

Question: How do we construct the *collection* of containers?

Answer:

Let $\mathscr{C} = \{C(I) : I \text{ is an independent set in } G\}$

Graph Container Lemma - Restated

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-$\textsc{IndepSet}$. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

[Kleitman, Winston '82]

Question: How do we construct the *collection* of containers?

Answer:

$$\begin{aligned}\text{Let } \mathscr{C} &= \{C(I) : I \text{ is an independent set in } G\} \\ &= \{C(F(I)) : I \text{ is an indepedent set in } G\}\end{aligned}$$

Graph Container Lemma - Restated

Lemma. Let $G = (V, E)$ be ϵ-far from ρ-$\textsc{IndepSet}$. Then, there exists a collection $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim \left(\rho - \frac{\epsilon}{\rho}\right) n$, and
3. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$.

[Kleitman, Winston '82]

Question: How do we construct the *collection* of containers?

Answer:

$$\begin{aligned} \text{Let } \mathscr{C} &= \{C(I) : I \text{ is an independent set in } G\} \\ &= \{C(F(I)) : I \text{ is an indepedent set in } G\} \\ &= \{C(F) : F = F(I) \text{ for some independent set } I \text{ in } G\} \end{aligned}$$

Testing Independent Sets

A New Graph Container Lemma

Lemma. Let G be ϵ-far from having a ρn independent set. Then, the collection $\mathscr{F} = \{F(I) : I \text{ is an independent set in } G\}$ satisfies:

A New Graph Container Lemma

Lemma. Let G be ϵ-far from having a ρn independent set. Then, the collection $\mathscr{F} = \{F(I) : I \text{ is an independent set in } G\}$ satisfies:

1. for every independent set I in G, there exists $F \in \mathscr{F}$ such that $F \subseteq I \subseteq C(F)$, and
2. for every $F \in \mathscr{F}$, $|F| \lesssim \rho^2/\epsilon$ and $|C(F)| \lesssim \left(\rho - |F| \cdot \frac{\epsilon}{\rho}\right) n$.

A New Graph Container Lemma

Lemma. Let G be ϵ-far from having a ρn independent set. Then, the collection $\mathscr{F} = \{F(I) : I \text{ is an independent set in } G\}$ satisfies:

1. for every independent set I in G, there exists $F \in \mathscr{F}$ such that $F \subseteq I \subseteq C(F)$, and
2. for every $F \in \mathscr{F}$, $|F| \lesssim \rho^2/\epsilon$ and $|C(F)| \lesssim \left(\rho - |F| \cdot \frac{\epsilon}{\rho}\right) n$.

Large fingerprints give very small containers.

A New Graph Container Lemma

Lemma. Let G be ϵ-far from having a ρn independent set. Then, the collection $\mathscr{F} = \{F(I) : I \text{ is an independent set in } G\}$ satisfies:

1. for every independent set I in G, there exists $F \in \mathscr{F}$ such that $F \subseteq I \subseteq C(F)$, and
2. for every $F \in \mathscr{F}$, $|F| \lesssim \rho^2/\epsilon$ and $|C(F)| \lesssim \left(\rho - |F| \cdot \frac{\epsilon}{\rho}\right) n$.

Large fingerprints give very small containers.

For example, at most $\binom{n}{O(1)}$ containers of size roughly $\left(\rho - \frac{\epsilon}{\rho}\right) n$.

Proof of the Main Theorem

Theorem. $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{O}(\rho^3/\epsilon^2)$.

Proof of the Main Theorem

Theorem. $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{O}(\rho^3/\epsilon^2)$.

Proof: The tester samples a set S of $s = \widetilde{O}(\rho^3/\epsilon^2)$ vertices uniformly at random. It accepts if and only if $G[S]$ contains a ρs-independent set.

- Easy case: if G contains ρn independent set.
- Hard case: if G is ϵ-far from having ρn independent set.

Proof of the Main Theorem

Theorem. $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{O}(\rho^3/\epsilon^2)$.

Proof: The tester samples a set S of $s = \widetilde{O}(\rho^3/\epsilon^2)$ vertices uniformly at random. It accepts if and only if $G[S]$ contains a ρs-independent set.

- Easy case: if G contains ρn independent set.
- Hard case: if G is ϵ-far from having ρn independent set.
  - Union bound over the set of fingerprints/containers.

Proof of the Main Theorem

Theorem. $\mathcal{S}_{\rho\text{-IndepSet}}(n, \epsilon) = \widetilde{O}(\rho^3/\epsilon^2)$.

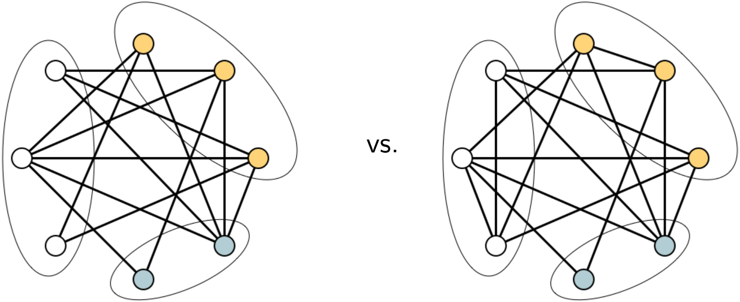
Proof: The tester samples a set S of $s = \widetilde{O}(\rho^3/\epsilon^2)$ vertices uniformly at random. It accepts if and only if $G[S]$ contains a ρs-independent set.

- Easy case: if G contains ρn independent set.
- Hard case: if G is ϵ-far from having ρn independent set.
  - Union bound over the set of fingerprints/containers.

$$
\begin{aligned}
&\Pr[S \text{ contains } \rho s \text{ independent set}] \\
&\le \sum_{F \in \mathscr{F}} \Pr[S \text{ contains a } \rho s \text{ independent set with fingerprint } F] \\
&\le \sum_{F \in \mathscr{F}} \Pr[S \text{ contains } F \text{ and at least } \rho s \text{ vertices from } C(F)].
\end{aligned}
$$

Testing Colorability

Testing k-Colorability



Goal: Distinguish between

(i) G is k-colorable, and

(ii) G is ϵ-far from being k-colorable.

Results: Testing k-Colorability

Prior Results. The sample complexity $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon)$ for testing k-colorability is bounded above by:

- $O(k^2/\epsilon^3)$ [Goldreich, Goldwasser, Ron '98]
- $O(k/\epsilon^2)$ [Alon, Krivelevich '02]
- $O(k^4/\epsilon)$ [Sohler '12]

and it is bounded below by

- $\widetilde{\Omega}(1/\epsilon)$ [Alon, Krivelevich '02]

Results: Testing k-Colorability

Prior Results. The sample complexity $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon)$ for testing k-colorability is bounded above by:

- $O(k^2/\epsilon^3)$ [Goldreich, Goldwasser, Ron '98]
- $O(k/\epsilon^2)$ [Alon, Krivelevich '02]
- $O(k^4/\epsilon)$ [Sohler '12]

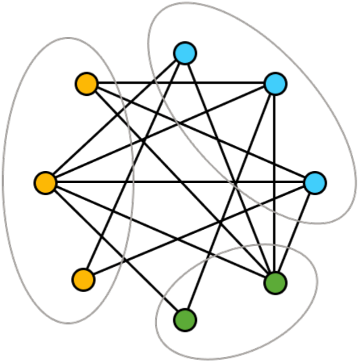
and it is bounded below by

- $\widetilde{\Omega}(1/\epsilon)$ [Alon, Krivelevich '02]

Theorem. $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon) = \widetilde{O}(k/\epsilon)$.

Container Method for k-Colorability

Any k-colorable subgraph can be partitioned into k disjoint independent sets.



Run container procedure on each independent set, and take the union of the containers.

Summary: Property Testing and the Container Method

New graph container lemmas can be used to show:

- $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{\Theta}(\rho^3/\epsilon^2)$
- $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon) = \widetilde{O}(k/\epsilon)$

Summary: Property Testing and the Container Method

New graph container lemmas can be used to show:

- $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{\Theta}(\rho^3/\epsilon^2)$
- $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon) = \widetilde{O}(k/\epsilon)$
- New sample complexity bound of $\widetilde{O}(kq^3/\epsilon)$ for satisfiability testing and new query complexity bound of $o((\rho^3/\epsilon^2)^2)$ for independent set testing.

[Blais, Seth, to appear at STOC '24]

Summary: Property Testing and the Container Method

New graph container lemmas can be used to show:

- $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{\Theta}(\rho^3/\epsilon^2)$
- $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon) = \widetilde{O}(k/\epsilon)$
- New sample complexity bound of $\widetilde{O}(kq^3/\epsilon)$ for satisfiability testing and new query complexity bound of $o((\rho^3/\epsilon^2)^2)$ for independent set testing.

[Blais, Seth, to appear at STOC '24]

What else can the graph container method give for property testing/sublinear algorithms?

Summary: Property Testing and the Container Method

New graph container lemmas can be used to show:

- $\mathcal{S}_{\rho\text{-}\textsc{IndepSet}}(n, \epsilon) = \widetilde{\Theta}(\rho^3/\epsilon^2)$
- $\mathcal{S}_{k\text{-}\textsc{Color}}(n, \epsilon) = \widetilde{O}(k/\epsilon)$
- New sample complexity bound of $\widetilde{O}(kq^3/\epsilon)$ for satisfiability testing and new query complexity bound of $o((\rho^3/\epsilon^2)^2)$ for independent set testing.

[Blais, Seth, to appear at STOC '24]

What else can the graph container method give for property testing/sublinear algorithms?

Thank you!