An Introduction to Hypergraph Containers

Sam Spiro, UC San Diego.

Based off of lecture notes by Balogh

Independent Sets of Hypergraphs

Recall that a hypergraph H is a pair (V, E) where $V = V(H)$ is a set (the vertex set) and $E = E(H)$ is a subsets of 2^V (the edge set).

Independent Sets of Hypergraphs

Recall that a hypergraph H is a pair (V, E) where $V = V(H)$ is a set (the vertex set) and $E = E(H)$ is a subsets of 2^V (the edge set).

An independent set I is a subset of V such that for all $e \in E$, $e \not\subset I$.

Independent Sets of Hypergraphs

Recall that a hypergraph H is a pair (V, E) where $V = V(H)$ is a set (the vertex set) and $E = E(H)$ is a subsets of 2^V (the edge set).

An independent set I is a subset of V such that for all $e \in E$, $e \not\subset I$. We let $\mathcal{I}(H)$ denote the set of independent sets of H and we let $\alpha(H)$ denote the size of a largest independent set of H.

Independent Sets of Hypergraphs

Recall that a hypergraph H is a pair (V, E) where $V = V(H)$ is a set (the vertex set) and $E = E(H)$ is a subsets of 2^V (the edge set).

An independent set I is a subset of V such that for all $e \in E$, $e \not\subset I$. We let $\mathcal{I}(H)$ denote the set of independent sets of H and we let $\alpha(H)$ denote the size of a largest independent set of H. Many problems in combinatorics can be phrased in terms of independent sets of hypergraphs.

Independent Sets of Hypergraphs

For example, let H_n^{Δ} be the hypergraph with $V = E(K_n)$ and E consisting of all triples $\{e, f, g\} \subseteq E(K_n)$ which form a triangle.

Independent Sets of Hypergraphs

For example, let H_n^{Δ} be the hypergraph with $V = E(K_n)$ and E consisting of all triples $\{e, f, g\} \subseteq E(K_n)$ which form a triangle. With this, independent sets $\mathcal{I}(H_n^{\Delta})$ correspond to triangle-free subgraphs of K_n.

Independent Sets of Hypergraphs

For example, let H_n^Δ be the hypergraph with $V = E(K_n)$ and E consisting of all triples $\{e, f, g\} \subseteq E(K_n)$ which form a triangle. With this, independent sets $\mathcal{I}(H_n^\Delta)$ correspond to triangle-free subgraphs of K_n. In particular,

Theorem (Mantel 1907)

$$\alpha(H_n^\Delta) = \left\lfloor n^2/4 \right\rfloor .$$

Independent Sets of Hypergraphs

As another example, define $H_{n,k}^{AP}$ by $V = [n] := \{1, \dots, n\}$ and $\{i_1, \dots, i_k\} \in E$ if this set forms a k-term arithmetic progression.

Independent Sets of Hypergraphs

As another example, define $H_{n,k}^{AP}$ by $V = [n] := \{1, \dots, n\}$ and $\{i_1, \dots, i_k\} \in E$ if this set forms a k-term arithmetic progression. Independent sets $\mathcal{I}(H_{n,k}^{AP})$ correspond to sets not containing any k-AP

Independent Sets of Hypergraphs

As another example, define $H_{n,k}^{AP}$ by $V = [n] := \{1, \ldots, n\}$ and $\{i_1, \ldots, i_k\} \in E$ if this set forms a k-term arithmetic progression. Independent sets $\mathcal{I}(H_{n,k}^{AP})$ correspond to sets not containing any k-AP, so in particular:

Theorem (Szemerédi 1975)

For any fixed k,

$$\alpha(H_{n,k}^{AP}) = o(n).$$

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$.

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$. Because many problems can be stated in terms of independent sets of hypergraphs, solving problems of this type has led to solutions to a large variety of problems:

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$.
Because many problems can be stated in terms of independent sets of hypergraphs, solving problems of this type has led to solutions to a large variety of problems:

- Extremal problems in random sets (Turán, Ramsey, Szemerédi).

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$. Because many problems can be stated in terms of independent sets of hypergraphs, solving problems of this type has led to solutions to a large variety of problems:

- Extremal problems in random sets (Turán, Ramsey, Szemerédi).
- Problems in discrete geometry.

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$. Because many problems can be stated in terms of independent sets of hypergraphs, solving problems of this type has led to solutions to a large variety of problems:

- Extremal problems in random sets (Turán, Ramsey, Szemerédi).
- Problems in discrete geometry.
- List coloring numbers of linear hypergraphs.

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$. Because many problems can be stated in terms of independent sets of hypergraphs, solving problems of this type has led to solutions to a large variety of problems:

- Extremal problems in random sets (Turán, Ramsey, Szemerédi).
- Problems in discrete geometry.
- List coloring numbers of linear hypergraphs.
- The KŁR conjecture.

Independent Sets of Hypergraphs

The main problem that we'll focus on is in estimating $|\mathcal{I}(H)|$. Because many problems can be stated in terms of independent sets of hypergraphs, solving problems of this type has led to solutions to a large variety of problems:

- Extremal problems in random sets (Turán, Ramsey, Szemerédi).
- Problems in discrete geometry.
- List coloring numbers of linear hypergraphs.
- The KŁR conjecture.
- $\cdots$

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$?

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)|$$

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H) \log |V(H)|}.$$

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H) \log |V(H)|}.$$

The upper bound follows because every independent is contained in a subset of size $\alpha(H)$.

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H) \log |V(H)|}.$$

The upper bound follows because every independent is contained in a subset of size $\alpha(H)$.

More generally, consider $\mathcal{C}$ a collection of subsets of vertices such that every independent set is a subset of some $C \in \mathcal{C}$.

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H) \log |V(H)|}.$$

The upper bound follows because every independent is contained in a subset of size $\alpha(H)$.

More generally, consider $\mathcal{C}$ a collection of subsets of vertices such that every independent set is a subset of some $C \in \mathcal{C}$. For example, $\mathcal{C} = \binom{V(H)}{\alpha(H)} := \{C \subseteq V(H) : |C| = \alpha(H)\}$ works.

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H) \log |V(H)|}.$$

The upper bound follows because every independent is contained in a subset of size $\alpha(H)$.

More generally, consider $\mathcal{C}$ a collection of subsets of vertices such that every independent set is a subset of some $C \in \mathcal{C}$. For example, $\mathcal{C} = \binom{V(H)}{\alpha(H)} := \{C \subseteq V(H) : |C| = \alpha(H)\}$ works. Then

$$|\mathcal{I}(H)| \leq \sum_{C \in \mathcal{C}} 2^{|C|}$$

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H)\log|V(H)|}.$$

The upper bound follows because every independent is contained in a subset of size $\alpha(H)$.

More generally, consider $\mathcal{C}$ a collection of subsets of vertices such that every independent set is a subset of some $C \in \mathcal{C}$. For example, $\mathcal{C} = \binom{V(H)}{\alpha(H)} := \{C \subseteq V(H) : |C| = \alpha(H)\}$ works. Then

$$|\mathcal{I}(H)| \leq \sum_{C \in \mathcal{C}} 2^{|C|} \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Independent Sets of Hypergraphs

Okay, but how do we bound $|\mathcal{I}(H)|$? Here are some trivial bounds:

$$2^{\alpha(H)} \leq |\mathcal{I}(H)| \leq \binom{|V(H)|}{\alpha(H)} 2^{\alpha(H)} \approx 2^{\alpha(H) \log |V(H)|}.$$

The upper bound follows because every independent is contained in a subset of size $\alpha(H)$.

More generally, consider $\mathcal{C}$ a collection of subsets of vertices such that every independent set is a subset of some $C \in \mathcal{C}$. For example, $\mathcal{C} = \binom{V(H)}{\alpha(H)} := \{C \subseteq V(H) : |C| = \alpha(H)\}$ works. Then

$$|\mathcal{I}(H)| \leq \sum_{C \in \mathcal{C}} 2^{|C|} \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

We call such a collection $\mathcal{C}$ a set of containers.

Independent Sets of Hypergraphs

If $\mathcal{C}$ is a collection of sets containing every independent set,

$$|\mathcal{I}(H)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Independent Sets of Hypergraphs

If $\mathcal{C}$ is a collection of sets containing every independent set,

$$|\mathcal{I}(H)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Thus if we can somehow find a set of containers $\mathcal{C}$ such that $|\mathcal{C}|$ and $|C|$ is small for all $C \in \mathcal{C}$, then we'll be able to get a significant improvement to the trivial upper bound of $|\mathcal{I}(H)|$.

Independent Sets of Hypergraphs

If $\mathcal{C}$ is a collection of sets containing every independent set,

$$|\mathcal{I}(H)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Thus if we can somehow find a set of containers $\mathcal{C}$ such that $|\mathcal{C}|$ and $|C|$ is small for all $C \in \mathcal{C}$, then we'll be able to get a significant improvement to the trivial upper bound of $|\mathcal{I}(H)|$. Note that it's very easy to do one or the other: take $\mathcal{C} = \binom{V(H)}{\alpha(H)}$ or $\mathcal{C} = \{V(H)\}$.

Independent Sets of Hypergraphs

If $\mathcal{C}$ is a collection of sets containing every independent set,

$$|\mathcal{I}(H)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Thus if we can somehow find a set of containers $\mathcal{C}$ such that $|\mathcal{C}|$ and $|C|$ is small for all $C \in \mathcal{C}$, then we'll be able to get a significant improvement to the trivial upper bound of $|\mathcal{I}(H)|$. Note that it's very easy to do one or the other: take $\mathcal{C} = \binom{V(H)}{\alpha(H)}$ or $\mathcal{C} = \{V(H)\}$.

The method of hypergraph containers gives a systematic way of finding a collection $\mathcal{C}$ with $|\mathcal{C}|$ and $|C|$ small.

Independent Sets of Hypergraphs

If $\mathcal{C}$ is a collection of sets containing every independent set,

$$|\mathcal{I}(H)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Thus if we can somehow find a set of containers $\mathcal{C}$ such that $|\mathcal{C}|$ and $|C|$ is small for all $C \in \mathcal{C}$, then we'll be able to get a significant improvement to the trivial upper bound of $|\mathcal{I}(H)|$. Note that it's very easy to do one or the other: take $\mathcal{C} = \binom{V(H)}{\alpha(H)}$ or $\mathcal{C} = \{V(H)\}$.

The method of hypergraph containers gives a systematic way of finding a collection $\mathcal{C}$ with $|\mathcal{C}|$ and $|C|$ small. Let's first try and get a handle of how this works for graphs.

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$.

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

The proof of this lemma will be algorithmic

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

The proof of this lemma will be algorithmic: we will input an independent set I and output a set of vertices $S = S(I)$.

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

The proof of this lemma will be algorithmic: we will input an independent set I and output a set of vertices $S = S(I)$.This will be done in such a way so that (1) S corresponds to some set $C = C(S)$ with $I \subseteq C$

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

The proof of this lemma will be algorithmic: we will input an independent set I and output a set of vertices $S = S(I)$.This will be done in such a way so that (1) S corresponds to some set $C = C(S)$ with $I \subseteq C$, (2) $\Delta(G[C]) < t - 1$

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

The proof of this lemma will be algorithmic: we will input an independent set I and output a set of vertices $S = S(I)$.This will be done in such a way so that (1) S corresponds to some set $C = C(S)$ with $I \subseteq C$, (2) $\Delta(G[C]) < t - 1$, and (3) $|S| \leq n/t$.

Graph Containers

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

The proof of this lemma will be algorithmic: we will input an independent set I and output a set of vertices $S = S(I)$.This will be done in such a way so that (1) S corresponds to some set $C = C(S)$ with $I \subseteq C$, (2) $\Delta(G[C]) < t - 1$, and (3) $|S| \leq n/t$. If we can do all this, then the lemma follows relatively easily.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v)$

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v) := C$.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v) := C$. If $d(v)$ is large then $|C|$ is relatively small.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v) := C$. If $d(v)$ is large then $|C|$ is relatively small.

In fact, we can improve upon this by taking

$$C = V(G) \setminus (N(v) \cup \{u \in V(G) : d(u) > d(v)\})$$

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v) := C$. If $d(v)$ is large then $|C|$ is relatively small.

In fact, we can improve upon this by taking

$$C = V(G) \setminus (N(v) \cup \{u \in V(G) : d(u) > d(v)\}),$$

since v is the vertex of largest degree in I.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v) := C$. If $d(v)$ is large then $|C|$ is relatively small.

In fact, we can improve upon this by taking

$$C = V(G) \setminus (N(v) \cup \{u \in V(G) : d(u) > d(v)\}),$$

since v is the vertex of largest degree in I. Thus even if $d(v)$ is small we can remove a lot of vertices.

Graph Containers

Goal: input I, output small S which corresponds to small $C \supset I$.

Say we wanted to do this by letting S be a single vertex of I. In this case, a reasonable choice is to let $S = \{v\}$ where $d(v) = \max_{u \in I} d(u)$. Indeed, given $S = \{v\}$ we have $I \subseteq V(G) \setminus N(v) := C$. If $d(v)$ is large then $|C|$ is relatively small.

In fact, we can improve upon this by taking

$$C = V(G) \setminus (N(v) \cup \{u \in V(G) : d(u) > d(v)\}),$$

since v is the vertex of largest degree in I. Thus even if $d(v)$ is small we can remove a lot of vertices. Our actual algorithm is essentially just an iterated version of this idea.

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices).

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$.

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$. Otherwise:

Step 2. Let $v \in A$ be the vertex with maximum degree in $G[A]$ (with ties broken according to some arbitrary ordering of $V(G)$).

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$. Otherwise:

Step 2. Let $v \in A$ be the vertex with maximum degree in $G[A]$ (with ties broken according to some arbitrary ordering of $V(G)$). If $v \notin I$, then set $A = A \setminus \{v\}$ and repeat Step 1.

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$. Otherwise:

Step 2. Let $v \in A$ be the vertex with maximum degree in $G[A]$ (with ties broken according to some arbitrary ordering of $V(G)$). If $v \notin I$, then set $A = A \setminus \{v\}$ and repeat Step 1. If $v \in I$:

Step 3. Set $S = S \cup \{v\}$ and $A = A \setminus (\{v\} \cup N_{G[A]}(v))$.

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$. Otherwise:

Step 2. Let $v \in A$ be the vertex with maximum degree in $G[A]$ (with ties broken according to some arbitrary ordering of $V(G)$). If $v \notin I$, then set $A = A \setminus \{v\}$ and repeat Step 1. If $v \in I$:

Step 3. Set $S = S \cup \{v\}$ and $A = A \setminus (\{v\} \cup N_{G[A]}(v))$. Repeat Step 1. $\square$

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$. Otherwise:

Step 2. Let $v \in A$ be the vertex with maximum degree in $G[A]$ (with ties broken according to some arbitrary ordering of $V(G)$). If $v \notin I$, then set $A = A \setminus \{v\}$ and repeat Step 1. If $v \in I$:

Step 3. Set $S = S \cup \{v\}$ and $A = A \setminus (\{v\} \cup N_{G[A]}(v))$. Repeat Step 1. $\square$

By construction $I \subseteq S \cup A = C(S)$,

Graph Containers

Input an independent set I. Start with $S = \emptyset$ (the selected vertices) and $A = V(G)$ (the available vertices). Iteratively proceed as follows:

Step 1. If $\Delta(G[A]) < t - 1$, output $S(I) = S$ and set $C(S) = S \cup A$. Otherwise:

Step 2. Let $v \in A$ be the vertex with maximum degree in $G[A]$ (with ties broken according to some arbitrary ordering of $V(G)$). If $v \notin I$, then set $A = A \setminus \{v\}$ and repeat Step 1. If $v \in I$:

Step 3. Set $S = S \cup \{v\}$ and $A = A \setminus (\{v\} \cup N_{G[A]}(v))$. Repeat Step 1. $\square$

By construction $I \subseteq S \cup A = C(S)$, and $\Delta(G[C(S)]) < t - 1$.

Graph Containers

A priori the set $C(S)$ is not well defined because it's defined in terms of S and another set A.

Graph Containers

A priori the set $C(S)$ is not well defined because it's defined in terms of S and another set A.

Claim

If S is the output of the previous algorithm, then S uniquely determines what A must have been at the end, i.e. $C(S)$ is well defined.

Graph Containers

A priori the set $C(S)$ is not well defined because it's defined in terms of S and another set A.

Claim

If S is the output of the previous algorithm, then S uniquely determines what A must have been at the end, i.e. $C(S)$ is well defined.

Accepting this, we define

$$\mathcal{C} = \{C(S(I)) : I \in \mathcal{I}(H).\}$$

Graph Containers

A priori the set $C(S)$ is not well defined because it's defined in terms of S and another set A.

Claim

If S is the output of the previous algorithm, then S uniquely determines what A must have been at the end, i.e. $C(S)$ is well defined.

Accepting this, we define

$$\mathcal{C} = \{C(S(I)) : I \in \mathcal{I}(H).\}$$

As noted on the previous slide, this is a set of containers which induce graphs with small maximum degree.

Graph Containers

A priori the set $C(S)$ is not well defined because it's defined in terms of S and another set A.

Claim

If S is the output of the previous algorithm, then S uniquely determines what A must have been at the end, i.e. $C(S)$ is well defined.

Accepting this, we define

$$\mathcal{C} = \{C(S(I)) : I \in \mathcal{I}(H).\}$$

As noted on the previous slide, this is a set of containers which induce graphs with small maximum degree. The last thing we need to prove is that $|\mathcal{C}|$ is small, and it suffices to show that $|\{S(I) : I \in \mathcal{I}(H)\}|$ is small.

Graph Containers

Lemma

If $I \in \mathcal{I}(H)$, *then* $|S(I)| \leq n/t$.

Graph Containers

Lemma

If $I \in \mathcal{I}(H)$, then $|S(I)| \leq n/t$.

If the algorithm is running then $\Delta(G[A]) \geq t - 1$.

Graph Containers

Lemma

If $I \in \mathcal{I}(H)$, then $|S(I)| \leq n/t$.

If the algorithm is running then $\Delta(G[A]) \geq t-1$. Thus every time we add a vertex v to S, we delete from A the vertex v and all of its (at least $t-1$) neighbors.

Graph Containers

Lemma

If $I \in \mathcal{I}(H)$, then $|S(I)| \leq n/t$.

If the algorithm is running then $\Delta(G[A]) \geq t - 1$. Thus every time we add a vertex v to S, we delete from A the vertex v and all of its (at least $t - 1$) neighbors. Because A has n vertices to start and we remove at least t vertices each time an element is added to S, we must have $|S| \leq n/t$. □

Graph Containers

Lemma

If $I \in \mathcal{I}(H)$, then $|S(I)| \leq n/t$.

If the algorithm is running then $\Delta(G[A]) \geq t-1$. Thus every time we add a vertex v to S, we delete from A the vertex v and all of its (at least $t-1$) neighbors. Because A has n vertices to start and we remove at least t vertices each time an element is added to S, we must have $|S| \leq n/t$. □

In total we have

$$|\mathcal{C}| \leq |\{S(I) : I \in \mathcal{I}(H)\}| \leq \sum_{k \leq n/t} \binom{n}{k},$$

giving the desired result. □

d-regular Graphs

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

d-regular Graphs

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

Our main application of this lemma is to effectively bound $|\mathcal{I}(G)|$ when G is a d-regular graph.

d-regular Graphs

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

Our main application of this lemma is to effectively bound $|\mathcal{I}(G)|$ when G is a d-regular graph. As a test case, if G is $n/2d$ disjoint copies of $K_{d,d}$, then

$$|\mathcal{I}(G)| = (2^{d+1} - 1)^{n/2d}$$

d-regular Graphs

Lemma

Let G be a graph on n vertices and $t \in \mathbb{R}$. There is a collection $\mathcal{C}$ of containers such that

(a) *For all $C \in \mathcal{C}$, we have $\Delta(G[C]) < t - 1$.*

(b) $|\mathcal{C}| \leq \sum_{k \leq n/t} \binom{n}{k} \approx \binom{n}{n/t}$.

Our main application of this lemma is to effectively bound $|\mathcal{I}(G)|$ when G is a d-regular graph. As a test case, if G is $n/2d$ disjoint copies of $K_{d,d}$, then

$$|\mathcal{I}(G)| = (2^{d+1} - 1)^{n/2d} = 2^{n/2+o(n)} \quad \text{if } d = \omega(1).$$

We'll show that this bound is asymptotically tight (for a reasonable range of d).

d-regular Graphs

To prove this, we start with a d-regular graph G and apply the container lemma (for some t) to get a set of containers $\mathcal{C}$.

d-regular Graphs

To prove this, we start with a d-regular graph G and apply the container lemma (for some t) to get a set of containers $\mathcal{C}$. As noted before, we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

d-regular Graphs

To prove this, we start with a d-regular graph G and apply the container lemma (for some t) to get a set of containers $\mathcal{C}$. As noted before, we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

However, our lemma tells us nothing about $|C|$, only that it has small maximum degree.

d-regular Graphs

To prove this, we start with a d-regular graph G and apply the container lemma (for some t) to get a set of containers $\mathcal{C}$. As noted before, we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

However, our lemma tells us nothing about $|C|$, only that it has small maximum degree. Thus we need to show that every induced subgraph of a d-regular graph with small maximum degree is small.

d-regular Graphs

Lemma

If G is an n-vertex d-regular graph with $C \subseteq V(G)$ and $|C| = n/2 + \epsilon n$, then $\Delta(G[C]) \geq \epsilon d$.

d-regular Graphs

Lemma

If G is an n-vertex d-regular graph with $C \subseteq V(G)$ and $|C| = n/2 + \epsilon n$, then $\Delta(G[C]) \geq \epsilon d$.

Note that taking $|C| > n/2$ is a natural choice since the disjoint union of $K_{d,d}$'s shows we can have independent sets of size $n/2$.

d-regular Graphs

Lemma

If G is an n-vertex d-regular graph with $C \subseteq V(G)$ and $|C| = n/2 + \epsilon n$, then $\Delta(G[C]) \geq \epsilon d$.

Note that taking $|C| > n/2$ is a natural choice since the disjoint union of $K_{d,d}$'s shows we can have independent sets of size $n/2$. In other words, this is a supersaturation result saying that if G is a d-regular graph, then (morally speaking) any set of size $(1+\epsilon)\alpha(G)$ contains many edges.

d-regular Graphs

Lemma

If G is an n-vertex d-regular graph with $C \subseteq V(G)$ and $|C| = n/2 + \epsilon n$, then $\Delta(G[C]) \geq \epsilon d$.

Note that taking $|C| > n/2$ is a natural choice since the disjoint union of $K_{d,d}$'s shows we can have independent sets of size $n/2$. In other words, this is a supersaturation result saying that if G is a d-regular graph, then (morally speaking) any set of size $(1 + \epsilon)\alpha(G)$ contains many edges.

In general to use the method of containers one often needs some form of supersaturation.

d-regular Graphs

Recall that we're trying to bound $|\mathcal{I}(G)|$ for a d-regular graph G, and that for $\mathcal{C}$ a set of containers we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

d-regular Graphs

Recall that we're trying to bound $|\mathcal{I}(G)|$ for a d-regular graph G, and that for $\mathcal{C}$ a set of containers we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Using $\mathcal{C}$ from our container lemma gives

$$|\mathcal{C}| \approx \binom{n}{n/t} \approx 2^{\frac{n}{t} \log n}$$

and that each C has maximum degree at most t

d-regular Graphs

Recall that we're trying to bound $|\mathcal{I}(G)|$ for a d-regular graph G, and that for $\mathcal{C}$ a set of containers we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Using $\mathcal{C}$ from our container lemma gives

$$|\mathcal{C}| \approx \binom{n}{n/t} \approx 2^{\frac{n}{t} \log n}$$

and that each C has maximum degree at most t, so by the previous lemma we have

$$2^{|C|} \leq 2^{n/2 + \frac{t}{d} n}.$$

d-regular Graphs

Recall that we're trying to bound $|\mathcal{I}(G)|$ for a d-regular graph G, and that for $\mathcal{C}$ a set of containers we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Using $\mathcal{C}$ from our container lemma gives

$$|\mathcal{C}| \approx \binom{n}{n/t} \approx 2^{\frac{n}{t} \log n}$$

and that each C has maximum degree at most t, so by the previous lemma we have

$$2^{|C|} \leq 2^{n/2 + \frac{t}{d} n}.$$

These bounds are optimized when $2^{\frac{n}{t} \log n} = 2^{\frac{t}{d} n}$, i.e. when $t = \sqrt{d \log n}$.

d-regular Graphs

Recall that we're trying to bound $|\mathcal{I}(G)|$ for a d-regular graph G, and that for $\mathcal{C}$ a set of containers we have

$$|\mathcal{I}(G)| \leq |\mathcal{C}| \cdot 2^{\max_{C \in \mathcal{C}} |C|}.$$

Using $\mathcal{C}$ from our container lemma gives

$$|\mathcal{C}| \approx \binom{n}{n/t} \approx 2^{\frac{n}{t} \log n}$$

and that each C has maximum degree at most t, so by the previous lemma we have

$$2^{|C|} \leq 2^{n/2 + \frac{t}{d} n}.$$

These bounds are optimized when $2^{\frac{n}{t} \log n} = 2^{\frac{t}{d} n}$, i.e. when $t = \sqrt{d \log n}$. In total this gives an upper bound of

$$2^{n/2 + 2\sqrt{\frac{\log n}{d}} n},$$

and this equals $2^{n/2 + o(n)}$ provided $d = \omega(\log n)$.

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

- The number of q-colorings of a d-regular graph.

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

- The number of q-colorings of a d-regular graph.
- A random version of Sperner's theorem.

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

- The number of q-colorings of a d-regular graph.
- A random version of Sperner's theorem.
- Counting the number of Sidon sets.

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

- The number of q-colorings of a d-regular graph.
- A random version of Sperner's theorem.
- Counting the number of Sidon sets.
- Counting the number of C_4-free graphs.

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

- The number of q-colorings of a d-regular graph.
- A random version of Sperner's theorem.
- Counting the number of Sidon sets.
- Counting the number of C_4-free graphs.

In each of these cases one applies the graph container lemma (or one of its many variants) in conjunction with a supersaturation lemma.

Hypergraph Containers

With graph containers alone one can solve a number of interesting problems such as:

- The number of q-colorings of a d-regular graph.
- A random version of Sperner's theorem.
- Counting the number of Sidon sets.
- Counting the number of C_4-free graphs.

In each of these cases one applies the graph container lemma (or one of its many variants) in conjunction with a supersaturation lemma.

However, to count e.g. $K_{s,t}$-free graphs, the most natural setting is to consider independent sets of hypergraphs.

Hypergraph Containers

There are many different variants of the container lemma for hypergraphs, which is both a benefit and an annoyance.

Hypergraph Containers

There are many different variants of the container lemma for hypergraphs, which is both a benefit and an annoyance. The key requirement for all of them is that the relevant hypergraph H must be "spread out."

Hypergraph Containers

There are many different variants of the container lemma for hypergraphs, which is both a benefit and an annoyance. The key requirement for all of them is that the relevant hypergraph H must be "spread out."

In particular, we require the maximum degree of H to be close to its average degree, and for its codegrees to be relatively small.

Hypergraph Containers

There are many different variants of the container lemma for hypergraphs, which is both a benefit and an annoyance. The key requirement for all of them is that the relevant hypergraph H must be "spread out."

In particular, we require the maximum degree of H to be close to its average degree, and for its codegrees to be relatively small.

To state this precisely, for $S \subseteq V(H)$ we define $d(S)$ to be the number of edges in H containing S.

Hypergraph Containers

There are many different variants of the container lemma for hypergraphs, which is both a benefit and an annoyance. The key requirement for all of them is that the relevant hypergraph H must be "spread out."

In particular, we require the maximum degree of H to be close to its average degree, and for its codegrees to be relatively small.

To state this precisely, for $S \subseteq V(H)$ we define $d(S)$ to be the number of edges in H containing S. We define

$$\Delta_i(H) = \max_{S \subseteq V(H): |S| = i} d(S)$$

to be the maximum i-degree of H.

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

For r a positive integer, there exists a constant $\delta > 0$ such that the following holds. Assume H is an r-uniform hypergraph such that

$$\Delta_i(H) \leq q^{i-1}\frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$. Then there exists a collection of containers $\mathcal{C}$ such that

(a) $|\mathcal{C}| \leq \sum_{k \leq qr|V(H)|} \binom{|V(H)|}{k}$,

(b) $|C| \leq (1 - \delta/c)|V(H)|$ *for all* $C \in \mathcal{C}$.

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

If

$$\Delta_i(H) \leq q^{i-1} \frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$, then there exists a set of containers $\mathcal{C}$ with

(a)

$$|\mathcal{C}| \approx \binom{|V(H)|}{qr|V(H)|}.$$

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

If

$$\Delta_i(H) \le q^{i-1} \frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$, then there exists a set of containers $\mathcal{C}$ with

(a)

$$|\mathcal{C}| \approx \binom{|V(H)|}{qr|V(H)|}.$$

If $i = 1$, then the hypothesis of the theorem requires $\Delta_1(H) \le c|E(H)|/|V(H)|$, i.e. the max degree is close to the average degree.

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

If

$$\Delta_i(H) \leq q^{i-1} \frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$, then there exists a set of containers $\mathcal{C}$ with

(a)

$$|\mathcal{C}| \approx \binom{|V(H)|}{qr|V(H)|}.$$

If $i = 1$, then the hypothesis of the theorem requires $\Delta_1(H) \leq c|E(H)|/|V(H)|$, i.e. the max degree is close to the average degree. In general, $\Delta_i(H)$ is at most q^{i-1} times the average degree of H

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

If

$$\Delta_i(H) \leq q^{i-1}\frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$, then there exists a set of containers $\mathcal{C}$ with

(a)

$$|\mathcal{C}| \approx \binom{|V(H)|}{qr|V(H)|}.$$

If $i = 1$, then the hypothesis of the theorem requires $\Delta_1(H) \leq c|E(H)|/|V(H)|$, i.e. the max degree is close to the average degree. In general, $\Delta_i(H)$ is at most q^{i-1} times the average degree of H, and the smaller q is (i.e. the more dispersed the edges of H are), the better control one has over $|\mathcal{C}|$.

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

There exist $\delta > 0$ *such that if*

$$\Delta_i(H) \leq q^{i-1} \frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$*, then there exists a set of containers* $\mathcal{C}$ *with*

(b) $|C| \leq (1 - \delta/c)|V(H)|$ *for all* $C \in \mathcal{C}$.

In general the bound of (b) for $|C|$ is too weak

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

There exist $\delta > 0$ *such that if*

$$\Delta_i(H) \leq q^{i-1} \frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$*, then there exists a set of containers* $\mathcal{C}$ *with*

(b) $|C| \leq (1 - \delta/c)|V(H)|$ *for all* $C \in \mathcal{C}$*.*

In general the bound of (b) for $|C|$ is too weak, and in this case one reapplies the container lemma to $H[C]$.

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

There exist $\delta > 0$ *such that if*

$$\Delta_i(H) \leq q^{i-1}\frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$*, then there exists a set of containers* $\mathcal{C}$ *with*

(b) $|C| \leq (1-\delta/c)|V(H)|$ *for all* $C \in \mathcal{C}$.

In general the bound of (b) for $|C|$ is too weak, and in this case one reapplies the container lemma to $H[C]$. Because $|C|$ is smaller than $|V(H)|$ for some constant factor, one only has to iterate this a few number of times

Hypergraph Containers

Theorem (Balogh-Morris-Samotij, Saxton-Thomason)

There exist $\delta > 0$ *such that if*

$$\Delta_i(H) \leq q^{i-1}\frac{c|E(H)|}{|V(H)|}$$

for all $i \in [r]$, *then there exists a set of containers* $\mathcal{C}$ *with*

(b) $|C| \leq (1 - \delta/c)|V(H)|$ *for all* $C \in \mathcal{C}$.

In general the bound of (b) for $|C|$ is too weak, and in this case one reapplies the container lemma to $H[C]$. Because $|C|$ is smaller than $|V(H)|$ for some constant factor, one only has to iterate this a few number of times (assuming H is "similar" to all its subgraphs $H[C]$).

Hypergraph Containers

Lastly, I'll note that many hypergraphs of interests satisfy the conditions of this container lemma.

Hypergraph Containers

Lastly, I'll note that many hypergraphs of interests satisfy the conditions of this container lemma. For example, this holds with H_n^{Δ} the hypergraph of triangles of K_n, and $H_{n,k}^{AP}$ the hypergraph of k-term arithmetic progressions.

Hypergraph Containers

Lastly, I'll note that many hypergraphs of interests satisfy the conditions of this container lemma. For example, this holds with H_n^{Δ} the hypergraph of triangles of K_n, and $H_{n,k}^{AP}$ the hypergraph of k-term arithmetic progressions.

Thus using the lemma and some (mostly routine) computations, one can immediately deduce nice results for problems involving these hypergraphs.

The End

Thank You!