Robustness of graph/hypergraph properties

Asaf Ferber (UCI)

Grad students seminar, UCI

May 1, 2020

Introduction

Recall that a k-uniform hypergraph (or a k-graph for short) $G = (V, E)$ consists of a set V of vertices, and a set E of edges, where $E \subseteq \binom{V}{k}$. The case $k = 2$ is referred to as a graph.

Introduction

Recall that a k-uniform hypergraph (or a k-graph for short) $G = (V, E)$ consists of a set V of vertices, and a set E of edges, where $E \subseteq \binom{V}{k}$. The case $k = 2$ is referred to as a graph.

A typical result in graph/hypergraph theory has the following flavor:

Theorem

Let $\mathcal{P}$ be a k-graph property. Then, every k-graph G that satisfies certain conditions has the property $\mathcal{P}$.

Introduction

Examples of interest:

Dirac's theorem (1952)

Every bipartite (2-)graph with both parts of size n and minimum degree at least $n/2$ contains a perfect matching.

Introduction

Examples of interest:

Dirac's theorem (1952)

Every bipartite (2-)graph with both parts of size n and minimum degree at least $n/2$ contains a perfect matching.

Dirac's theorem (1952)

Every graph on $n \geq 3$ vertices and with minimum degree at least $n/2$ contains a Hamilton cycle.

Introduction

Examples of interest:

Dirac's theorem (1952)

Every bipartite (2-)graph with both parts of size n and minimum degree at least $n/2$ contains a perfect matching.

Dirac's theorem (1952)

Every graph on $n \geq 3$ vertices and with minimum degree at least $n/2$ contains a Hamilton cycle.

Once such a result is established, it is natural to ask how strongly does G possess $\mathcal{P}$? In other words, we want to determine the robustness of G with respect to $\mathcal{P}$.

Measures of robustness

We illustrate various measures that can be used to study the robustness of graph properties. For example, suppose that G is any graph that contains a Hamilton cycle. Then, one could ask:

- How many Hamilton cycles it contains?

Measures of robustness

We illustrate various measures that can be used to study the robustness of graph properties. For example, suppose that G is any graph that contains a Hamilton cycle. Then, one could ask:

- How many Hamilton cycles it contains?
- What is the largest collection of edge-disjoint Hamilton cycles it has?

Measures of robustness

We illustrate various measures that can be used to study the robustness of graph properties. For example, suppose that G is any graph that contains a Hamilton cycle. Then, one could ask:

- How many Hamilton cycles it contains?
- What is the largest collection of edge-disjoint Hamilton cycles it has?
- Can Maker build a Hamilton cycle while playing on its edges?

Measures of robustness

We illustrate various measures that can be used to study the robustness of graph properties. For example, suppose that G is any graph that contains a Hamilton cycle. Then, one could ask:

- How many Hamilton cycles it contains?
- What is the largest collection of edge-disjoint Hamilton cycles it has?
- Can Maker build a Hamilton cycle while playing on its edges?
- How many Hamilton cycles "survive" if we take a random subgraph of G?

Measures of robustness

We illustrate various measures that can be used to study the robustness of graph properties. For example, suppose that G is any graph that contains a Hamilton cycle. Then, one could ask:

- How many Hamilton cycles it contains?
- What is the largest collection of edge-disjoint Hamilton cycles it has?
- Can Maker build a Hamilton cycle while playing on its edges?
- How many Hamilton cycles "survive" if we take a random subgraph of G?
- etc.

Measures of robustness

Another measure of how strongly a graph satisfies some property is the so-called resilience of a property.

Measures of robustness

Another measure of how strongly a graph satisfies some property is the so-called resilience of a property. A graph property is called monotone increasing if it is closed under the addition of edges.

Measures of robustness

Another measure of how strongly a graph satisfies some property is the so-called resilience of a property. A graph property is called monotone increasing if it is closed under the addition of edges. For a monotone increasing graph property, the resilience quantifies the robustness in terms of the number of edges one must delete from G, locally or globally, in order to destroy the property. Specifically,

Measures of robustness

Another measure of how strongly a graph satisfies some property is the so-called resilience of a property. A graph property is called

monotone increasing if it is closed under the addition of edges. For

a monotone increasing graph property, the resilience quantifies the robustness in terms of the number of edges one must delete from G, locally or globally, in order to destroy the property. Specifically,

Definition

Let G be a graph satisfying $\mathcal{P}$.

- Its resilience w.r.t $\mathcal{P}$ is the max $0 \leq \alpha \leq 1$ for which every subgraph with at least $(1-\alpha)|E(G)|$ edges satisfies $\mathcal{P}$.

Measures of robustness

Another measure of how strongly a graph satisfies some property is the so-called resilience of a property. A graph property is called monotone increasing if it is closed under the addition of edges. For a monotone increasing graph property, the resilience quantifies the robustness in terms of the number of edges one must delete from G, locally or globally, in order to destroy the property. Specifically,

Definition

Let G be a graph satisfying $\mathcal{P}$.

- Its resilience w.r.t $\mathcal{P}$ is the max $0 \leq \alpha \leq 1$ for which every subgraph with at least $(1-\alpha)|E(G)|$ edges satisfies $\mathcal{P}$.
- Its local resilience w.r.t $\mathcal{P}$ is the max α for which every subgraph H of G for which $d_H(v) \geq (1-\alpha)d_G(v)$ for all v satisfies $\mathcal{P}$.

Models of interest

Among other popular models, one could study "robustness" w.r.t to certain properties of the following model:

- Dirac k-graphs k-graphs satisfying some minimum degree constraints to ensure $\mathcal{P}$

Models of interest

Among other popular models, one could study "robustness" w.r.t to certain properties of the following model:

- Dirac k-graphs k-graphs satisfying some minimum degree constraints to ensure $\mathcal{P}$

- $H^{(k)}_{n,p}$ the binomial model

Models of interest

Among other popular models, one could study "robustness" w.r.t to certain properties of the following model:

- Dirac k-graphs k-graphs satisfying some minimum degree constraints to ensure $\mathcal{P}$
- $H_{n,p}^{(k)}$ the binomial model
- $G_{n,d}$ the random regular graph model

Models of interest

Among other popular models, one could study "robustness" w.r.t to certain properties of the following model:

- Dirac k-graphs k-graphs satisfying some minimum degree constraints to ensure $\mathcal{P}$
- $H_{n,p}^{(k)}$ the binomial model
- $G_{n,d}$ the random regular graph model
- (n, d, λ)-graphs – spectral expanders

Models of interest

Among other popular models, one could study "robustness" w.r.t to certain properties of the following model:

- Dirac k-graphs k-graphs satisfying some minimum degree constraints to ensure $\mathcal{P}$
- $H^{(k)}_{n,p}$ the binomial model
- $G_{n,d}$ the random regular graph model
- (n, d, λ)-graphs – spectral expanders
- Random/deterministic directed/oriented graphs

Many graphs – history

Given a graph G, let $h(G)$ denote the number of distinct Hamilton cycles in G.

Many graphs – history

Given a graph G, let $h(G)$ denote the number of distinct Hamilton cycles in G. By Dirac's we have $h(G) > 0$ for every G with $\delta(G) \geq |V|/2$.

Many graphs – history

Given a graph G, let $h(G)$ denote the number of distinct Hamilton cycles in G. By Dirac's we have $h(G) > 0$ for every G with $\delta(G) \geq |V|/2$.

Question How many Hamilton cycles does a Dirac's graph must contain?

Many graphs – history

Given a graph G, let $h(G)$ denote the number of distinct Hamilton cycles in G. By Dirac's we have $h(G) > 0$ for every G with $\delta(G) \geq |V|/2$.

Question How many Hamilton cycles does a Dirac's graph must contain?

- Sárközy, Selkow and Szemerédi 2003 showed $h(G) \geq c^n n!$ for some small c, and conjectured that $c \approx \frac{1}{2}$.

Many graphs – history

Given a graph G, let $h(G)$ denote the number of distinct Hamilton cycles in G. By Dirac's we have $h(G) > 0$ for every G with $\delta(G) \geq |V|/2$.

Question How many Hamilton cycles does a Dirac's graph must contain?

- Sárközy, Selkow and Szemerédi 2003 showed $h(G) \geq c^n n!$ for some small c, and conjectured that $c \approx \frac{1}{2}$.
- Cuckler and Kahn 2009 proved the stronger result that for every Dirac graph G on n vertices one has

$$h(G) \geq \left(\frac{\delta(G)}{e}\right)^n (1 - o(1))^n .$$

Many graphs – (relatively) new approach

The proof of Cuckler and Kahn is quite involved and uses a self-avoiding random walk on G, in which the next vertex is chosen from the yet unvisited neighbors of the current vertex according to a very cleverly chosen distribution.

Many graphs – (relatively) new approach

The proof of Cuckler and Kahn is quite involved and uses a self-avoiding random walk on G, in which the next vertex is chosen from the yet unvisited neighbors of the current vertex according to a very cleverly chosen distribution.

A different approach to this problem was proposed by F., Krivelevich and Sudakov 2017. This approach is based on the standard estimates for the permanent of a matrix.

Many graphs – (relatively) new approach

The proof of Cuckler and Kahn is quite involved and uses a self-avoiding random walk on G, in which the next vertex is chosen from the yet unvisited neighbors of the current vertex according to a very cleverly chosen distribution.

A different approach to this problem was proposed by F., Krivelevich and Sudakov 2017. This approach is based on the standard estimates for the permanent of a matrix.

The permanent of an $n \times n$ matrix A is defined as $per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} A_{i\sigma(i)}$.
A 0-1 matrix A is called r-regular if it contains exactly r 1's in every row and every column.

Many graphs – permanent estimates

Key tools in our approach are the following two estimates on $per(A)$.

Many graphs – permanent estimates

Key tools in our approach are the following two estimates on $per(A)$.

Van Der Waerden Conjecture (Egorychev '81 and Falikman '81)

If A is an $n \times n$, r-regular matrix, then $per(A) \geq r^n \frac{n!}{n^n}$.

Many graphs – permanent estimates

Key tools in our approach are the following two estimates on $per(A)$.

Van Der Waerden Conjecture (Egorychev '81 and Falikman '81)

If A is an $n \times n$, r-regular matrix, then $per(A) \geq r^n \frac{n!}{n^n}$.

Minc's Conjecture (Bregman '73)

If A is an $n \times n$ matrix, then $per(A) \leq \prod_{i=1}^{n} (r_i!)^{1/r_i}$, where r_i is the number of 1s in the ith row.

Many graphs – permanent estimates

Key tools in our approach are the following two estimates on $per(A)$.

Van Der Waerden Conjecture (Egorychev '81 and Falikman '81)

If A is an $n \times n$, r-regular matrix, then $per(A) \geq r^n \frac{n!}{n^n}$.

Minc's Conjecture (Bregman '73)

If A is an $n \times n$ matrix, then $per(A) \leq \prod_{i=1}^{n} (r_i!)^{1/r_i}$, where r_i is the number of 1s in the ith row.

By combining the above estimates we obtain that for an r-regular 0/1 matrix we have that $per(A) = (1 - o(1))^n (r/e)^n$.

Many graphs – permanent cont'd

Observe that if A is the bipartite adjacency matrix of a bipartite graph G, then $per(A) =$'the number of perfect matchings in G'.

Many graphs – permanent cont'd

Observe that if A is the bipartite adjacency matrix of a bipartite graph G, then $per(A) =$'the number of perfect matchings in G'. If

A is the adjacency matrix of G, then $per(A)=$'the number of 2-factors of G'.

Many graphs – permanent cont'd

Observe that if A is the bipartite adjacency matrix of a bipartite graph G, then $per(A) =$'the number of perfect matchings in G'. If

A is the adjacency matrix of G, then $per(A) =$'the number of 2-factors of G'. To illustrate our technique we sketch the proof of

the following statement, which gives a lower bound on the number of Hamilton cycles in a dense graph G in terms of $\mathrm{reg}(G)$, where $\mathrm{reg}(G)$ is the maximal even r for which G contains an r-factor.

Theorem (F., Krivelevich and Sudakov 2017)

Let G be a Dirac graph on n vertices. Then the number of Hamilton cycles in G is at least $(1 - o(1))^n \left(\frac{\mathrm{reg}(G)}{e}\right)^n$.

Many graphs

Proof

- Let $r = \mathrm{reg}(G)$ and let $H \subseteq G$ be an r-factor of G.

Many graphs

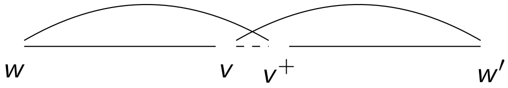
Proof

- Let $r = \mathrm{reg}(G)$ and let $H \subseteq G$ be an r-factor of G.
- By the above permanent estimate, one can show that H has at least $(r/e)^n(1 - o(1))^n$ 2-factors with at most $s = n^{1/2+o(1)}$ cycles in each.

Many graphs

Proof

- Let $r = \mathrm{reg}(G)$ and let $H \subseteq G$ be an r-factor of G.
- By the above permanent estimate, one can show that H has at least $(r/e)^n(1 - o(1))^n$ 2-factors with at most $s = n^{1/2+o(1)}$ cycles in each.
- For every such factor F, we can turn F into a Hamilton cycle of G by adding and removing $O(s)$ edges.



Many graphs

- In order to complete the proof, note that every Hamilton cycle C we constructed was counted at most $\binom{n}{s}(2s)^{2s} = r^{o(n)}$ times.

Many graphs

- In order to complete the proof, note that every Hamilton cycle C we constructed was counted at most $\binom{n}{s}(2s)^{2s} = r^{o(n)}$ times.
- Indeed, to get a 2-factor F from C, choose s edges of C to delete. This gives at most s paths which need to be turned into a 2-factor by connecting their endpoints.

Many graphs

- In order to complete the proof, note that every Hamilton cycle C we constructed was counted at most $\binom{n}{s}(2s)^{2s} = r^{o(n)}$ times.
- Indeed, to get a 2-factor F from C, choose s edges of C to delete. This gives at most s paths which need to be turned into a 2-factor by connecting their endpoints. For each endpoint we have at most $2s$ choices of other endpoints to connect it to.

Many graphs

- In order to complete the proof, note that every Hamilton cycle C we constructed was counted at most $\binom{n}{s}(2s)^{2s} = r^{o(n)}$ times.
- Indeed, to get a 2-factor F from C, choose s edges of C to delete. This gives at most s paths which need to be turned into a 2-factor by connecting their endpoints. For each endpoint we have at most $2s$ choices of other endpoints to connect it to.
- All in all, we obtain at least $(r/e)^n(1-o(1))^n$ Hamilton cycles.

Many graphs – other models

A tournament is an oriented complete graph. It is easy to prove by induction that every tournament has a Hamilton path. Moreover, if the tournament is regular, i.e., if all vertices have the same in/outdegrees, then it also has a Hamilton cycle.

Many graphs – other models

A tournament is an oriented complete graph. It is easy to prove by induction that every tournament has a Hamilton path. Moreover, if the tournament is regular, i.e., if all vertices have the same in/outdegrees, then it also has a Hamilton cycle. Counting

Hamilton cycles in tournaments goes back some seventy years to one of the first applications of the probabilistic method by Szele '43. He proved that there is a tournament on n vertices which has at least $(n-1)!/2^n$ Hamilton cycles.

Many graphs – other models

A tournament is an oriented complete graph. It is easy to prove by induction that every tournament has a Hamilton path. Moreover, if the tournament is regular, i.e., if all vertices have the same in/outdegrees, then it also has a Hamilton cycle. Counting

Hamilton cycles in tournaments goes back some seventy years to one of the first applications of the probabilistic method by Szele '43. He proved that there is a tournament on n vertices which has at least $(n-1)!/2^n$ Hamilton cycles.

Alon '90 showed, using permanent estimates, that every n-vertex tournament has at most $O(n^{3/2}(n-1)!/2^n)$ Hamilton cycles.

Many graphs – other models

A tournament is an oriented complete graph. It is easy to prove by induction that every tournament has a Hamilton path. Moreover, if the tournament is regular, i.e., if all vertices have the same in/outdegrees, then it also has a Hamilton cycle. Counting

Hamilton cycles in tournaments goes back some seventy years to one of the first applications of the probabilistic method by Szele '43. He proved that there is a tournament on n vertices which has at least $(n-1)!/2^n$ Hamilton cycles.

Alon '90 showed, using permanent estimates, that every n-vertex tournament has at most $O(n^{3/2}(n-1)!/2^n)$ Hamilton cycles.

Thomassen '85 conjectured that every regular tournament contains at least $n^{(1-o(1))n}$ Hamilton cycles.

Many graphs – other models

A tournament is an oriented complete graph. It is easy to prove by induction that every tournament has a Hamilton path. Moreover, if the tournament is regular, i.e., if all vertices have the same in/outdegrees, then it also has a Hamilton cycle. Counting

Hamilton cycles in tournaments goes back some seventy years to one of the first applications of the probabilistic method by Szele '43. He proved that there is a tournament on n vertices which has at least $(n-1)!/2^n$ Hamilton cycles.

Alon '90 showed, using permanent estimates, that every n-vertex tournament has at most $O(n^{3/2}(n-1)!/2^n)$ Hamilton cycles.

Thomassen '85 conjectured that every regular tournament contains at least $n^{(1-o(1))n}$ Hamilton cycles.

Cuckler 2007 proved, using the above mentioned random walk approach, that for every regular tournament T we have $h(T) \geq \frac{n!}{(2+o(1))^n}$.

Many graphs

Tournaments are a special case of oriented graphs. Given an oriented graph G, we let $\delta^{\pm}(G) = \min\{\delta^{+}(G), \delta^{-}(G)\}$ be the minimum semi-degree of G.

Many graphs

Tournaments are a special case of oriented graphs. Given an oriented graph G, we let $\delta^{\pm}(G) = \min\{\delta^{+}(G), \delta^{-}(G)\}$ be the minimum semi-degree of G.

Question (Tomassen, '70s) What is the minimal $\delta^{\pm}$ which implies Hamiltonicity?

Many graphs

Tournaments are a special case of oriented graphs. Given an oriented graph G, we let $\delta^{\pm}(G) = \min\{\delta^{+}(G), \delta^{-}(G)\}$ be the minimum semi-degree of G.

Question (Tomassen, '70s) What is the minimal $\delta^{\pm}$ which implies Hamiltonicity?

Häggkvist found a construction which gives a lower bound of $\frac{3n-4}{8} - 1$.

Many graphs

Tournaments are a special case of oriented graphs. Given an oriented graph G, we let $\delta^{\pm}(G) = \min\{\delta^{+}(G), \delta^{-}(G)\}$ be the minimum semi-degree of G.

Question (Tomassen, '70s) What is the minimal $\delta^{\pm}$ which implies Hamiltonicity?

Häggkvist found a construction which gives a lower bound of $\frac{3n-4}{8} - 1$.

Keevash, Kühn and Osthus 2009 proved that if $\delta^{\pm}(G) \geq \frac{3n-4}{8}$ then $h(G) > 0$ for large enough n.

Many graphs

Tournaments are a special case of oriented graphs. Given an oriented graph G, we let $\delta^{\pm}(G) = \min\{\delta^{+}(G), \delta^{-}(G)\}$ be the minimum semi-degree of G.

Question (Tomassen, '70s) What is the minimal $\delta^{\pm}$ which implies Hamiltonicity?

Häggkvist found a construction which gives a lower bound of $\frac{3n-4}{8} - 1$.

Keevash, Kühn and Osthus 2009 proved that if $\delta^{\pm}(G) \geq \frac{3n-4}{8}$ then $h(G) > 0$ for large enough n.

Using our permanent based approach we obtained

Theorem (F., Krievlevich and Sudakov '17)

Let G be an oriented graph with all in/out degrees in $cn \pm o(n)$ for some constant $c > 3/8$. Then $h(G) \geq \left(\frac{(c+o(1))n}{e}\right)^n$.

Many graphs – binomial model

In $G_{n,p}$, perhaps the most famous result is due independently to Pósa 1976 and Korshunov 1976, showing that whp it is Hamiltonian assuming that $p \geq C \log n/n$.

Many graphs – binomial model

In $G_{n,p}$, perhaps the most famous result is due independently to Pósa 1976 and Korshunov 1976, showing that whp it is Hamiltonian assuming that $p \geq C \log n/n$.

There were a number of subsequent improvements by different authors, leading to the sharp "hitting time" result of Bollobas 1984 and Ajtai, Komlós and Szemerédi 1985.

Many graphs – binomial model

In $G_{n,p}$, perhaps the most famous result is due independently to Pósa 1976 and Korshunov 1976, showing that whp it is Hamiltonian assuming that $p \geq C \log n/n$.

There were a number of subsequent improvements by different authors, leading to the sharp "hitting time" result of Bollobas 1984 and Ajtai, Komlós and Szemerédi 1985.

For the number of Hamilton cycles in $G_{n,p}$, observe that, by Markov's it is always at most $\omega(1)(n-1)!p^n$. It turns out that the lower bound is not that simple, and following results by Cooper and Frieze 1989, and Janson 1994, Glebov and Krivelevich 2013 obtained such a bound that holds for all p for which $G_{n,p}$ is typically Hamiltonian.

Many graphs – binomial model

A natural extension is to consider the same problem in the binomial directed graph $D_{n,p}$.

Many graphs – binomial model

A natural extension is to consider the same problem in the binomial directed graph $D_{n,p}$.

One of the first important insights, due to McDiarmid 1980, is that one can use a simple coupling argument to compare certain probabilities between $D_{n,p}$ and $G_{n,p}$. In particular, he showed that $\Pr[D_{n,p} \text{ is hamiltonian}] \geq \Pr[G_{n,p} \text{ is hamiltonian}]$.

Many graphs – binomial model

A natural extension is to consider the same problem in the binomial directed graph $D_{n,p}$.

One of the first important insights, due to McDiarmid 1980, is that one can use a simple coupling argument to compare certain probabilities between $D_{n,p}$ and $G_{n,p}$. In particular, he showed that $\Pr[D_{n,p} \text{ is hamiltonian}] \geq \Pr[G_{n,p} \text{ is hamiltonian}]$.

However, this is not optimal, and Frieze 1988 designed an algorithm to show Hamiltonicity at the optimal $p \geq (\log n + \omega(1))/n$.

McDiarmid's argument

In case that you are not familiar with such arguments, here we sketch McDiarmid's proof:

- Let $e_1, \ldots, e_N$ be an ordering of $E(K_n)$.

McDiarmid's argument

In case that you are not familiar with such arguments, here we sketch McDiarmid's proof:

- Let $e_1, \ldots, e_N$ be an ordering of $E(K_n)$.
- For $0 \leq i \leq N$, define Γ_i by: each e_j, $j < i$ is being added like in a $D_{n,p}$. Each $j \geq i$ is being added like in $G_{n,p}$.

McDiarmid's argument

In case that you are not familiar with such arguments, here we sketch McDiarmid's proof:

- Let $e_1, \ldots, e_N$ be an ordering of $E(K_n)$.
- For $0 \leq i \leq N$, define Γ_i by: each e_j, $j < i$ is being added like in a $D_{n,p}$. Each $j \geq i$ is being added like in $G_{n,p}$.
- Observe that $\Gamma_0 = G_{n,p}$ and $\Gamma_N = D_{n,p}$. Therefore, it is enough to show $\Pr[\Gamma_{i+1} \text{ hamiltonian}] \geq \Pr[\Gamma_i \text{ hamiltonian}]$.

McDiarmid's argument

In case that you are not familiar with such arguments, here we sketch McDiarmid's proof:

- Let $e_1, \ldots, e_N$ be an ordering of $E(K_n)$.
- For $0 \leq i \leq N$, define Γ_i by: each e_j, $j < i$ is being added like in a $D_{n,p}$. Each $j \geq i$ is being added like in $G_{n,p}$.
- Observe that $\Gamma_0 = G_{n,p}$ and $\Gamma_N = D_{n,p}$. Therefore, it is enough to show $\Pr[\Gamma_{i+1} \text{ hamiltonian}] \geq \Pr[\Gamma_i \text{ hamiltonian}]$.
- To see this, expose all arcs but those obtained from e_i. If Γ_{i-1} is Hamiltonian – done. If not hamiltonian even after adding e_i – done. If needs e_i, then at least one orientation is good and the probability to see it is at least p and we're done

Many graphs – binomial

For the counting part we again have an upper bound by Markov's. For the lower bound, improving upon F., Kronenberg and Long 2017 ($p \geq \log^2 /n$), F. and Long 2019 ($p \geq \log\log n \cdot \log n/n$), F. Kwan and Sudakov 2019 have finally proved an asymptotically tight lower bound that also works in a 'hitting time' version. Here we sketch a proof for $p = \omega(\log n/n)/$

Many graphs – binomial

For the counting part we again have an upper bound by Markov's. For the lower bound, improving upon F., Kronenberg and Long 2017 ($p \geq \log^2 /n$), F. and Long 2019 ($p \geq \log\log n \cdot \log n/n$), F. Kwan and Sudakov 2019 have finally proved an asymptotically tight lower bound that also works in a 'hitting time' version. Here we sketch a proof for $p = \omega(\log n/n)/$

Proof sketch (F., Kwan and Sudakov) The model $D_{n,p}$ can be coupled as follows:

- Let $p_1 \approx p$ and $p_2 \ll p$ be such that $1 - p = (1 - p_1)(1 - p_2)$.
- Expose a random bipartite graph B_{n,n,p_1}.
- Let $V_1 \cup V_2$ be its parts, and take a random labeling of $V_2 = \{u_1, \ldots, u_n\}$.
- Take D_{n,p_2}.
- Define $D := D_{n,p_2} \cup \{\overrightarrow{ij} \mid v_i u_j \in B_{n,n,p_1}\}$.

Many graphs

- It is not hard to show that B_{n,n,p_1} has the 'correct' number of perfect matchings using permanent estimates.

Many graphs

- It is not hard to show that B_{n,n,p_1} has the 'correct' number of perfect matchings using permanent estimates.
- Since each perfect matching corresponds to a permutation of V_1 into V_2, it follows that each such permutations is being mapped to a 1-factor in $D_{n,p}$ which corresponds to its cyclic form.

Many graphs

- It is not hard to show that B_{n,n,p_1} has the 'correct' number of perfect matchings using permanent estimates.
- Since each perfect matching corresponds to a permutation of V_1 into V_2, it follows that each such permutations is being mapped to a 1-factor in $D_{n,p}$ which corresponds to its cyclic form.
- Since the expected number of cycles in a random permutation is around $\log n$, by Markov's we know that 'most' permutations are mapped with at most (say) $\log^2 n$ many cycles.

Many graphs

- It is not hard to show that B_{n,n,p_1} has the 'correct' number of perfect matchings using permanent estimates.
- Since each perfect matching corresponds to a permutation of V_1 into V_2, it follows that each such permutations is being mapped to a 1-factor in $D_{n,p}$ which corresponds to its cyclic form.
- Since the expected number of cycles in a random permutation is around $\log n$, by Markov's we know that 'most' permutations are mapped with at most (say) $\log^2 n$ many cycles.
- Finally, given any such 1-factor, we show that whp, using the (random) edges of D_{n,p_2}, we can complete it into a Hamilton cycle.

This completes the proof.

Edge disjoint copies

A Hamilton decomposition of G is a collection of edge-disjoint Hamilton cycles covering all the edges of G.

Edge disjoint copies

A Hamilton decomposition of G is a collection of edge-disjoint Hamilton cycles covering all the edges of G.
For undirected graphs, Csaba, Kühn, Lo, Osthus and Treglown proved that every n-vertex Dirac graph G contains at least $\mathrm{reg}(n, \delta(G))/2$ edge-disjoint Hamilton cycles.

Edge disjoint copies

A Hamilton decomposition of G is a collection of edge-disjoint Hamilton cycles covering all the edges of G.
For undirected graphs, Csaba, Kühn, Lo, Osthus and Treglown proved that every n-vertex Dirac graph G contains at least $\mathrm{reg}(n, \delta(G))/2$ edge-disjoint Hamilton cycles.

Theorem (F., Krivelevich and Sudakov 2017)

Every n-vertex Dirac graph contains at least $(1 - o(1))reg(G)/2$ edge-disjoint Hamilton cycles.

For directed graphs, a remarkable result of Kühn and Osthus 2013, settling an old conjecture by Kelly, states that every r-regular oriented graph G has a Hamilton decomposition, provided that $r \geq 3n/8 + \varepsilon n$.

Edge disjoint copies

For non-regular, we conjecture:

Conjecture (F., Long and Sudakov 2017)

Let G be an oriented graph on n vertices with $\delta^{\pm}(G) \geq 3n/8$.
Then G contains $\mathrm{reg}(G)$ edge-disjoint Hamilton cycles.

Edge disjoint copies

For non-regular, we conjecture:

Conjecture (F., Long and Sudakov 2017)

Let G be an oriented graph on n vertices with $\delta^{\pm}(G) \geq 3n/8$. Then G contains $\mathrm{reg}(G)$ edge-disjoint Hamilton cycles.

Theorem (F., Long and Sudakov 2017)

If $\delta^{\pm}(G) \geq cn$ for some constant $c > 3/8$ then G contains $(1 - o(1))reg(G)$ edge-disjoint Hamilton cycles.

The proof also gives a rather short alternative proof of an approximate version of Kelly's conjecture.

Edge disjoint copies

- F., Kronenberg and Long 2017 studied counting and packing Hamilton cycles problems in random and pseudorandom (di)graphs.
- F. and Vu 2016 packing of perfect matchings in random hypergraphs (and introduced an 'online sprinkling technique').
- F. and Samotij 2018 packing bounded-degree trees in random graphs.
- F. and Jain 2019 spectral expanders have 1-factorization.
- F., Jain and Sudakov 2019 regular Dirac graphs have the 'correct' number of 1-factorizations.
- F. and Kwan 2020+ a random Steiner Triple System contains an approximate decomposition into perfect matchings.
- And more..

Edge disjoint copies – the packing problem

In a greater generality, one could state the following problem:

The Packing Problem

- Let $H_1, \ldots, H_N$ be a family of (hyper)graphs,

Edge disjoint copies – the packing problem

In a greater generality, one could state the following problem:

The Packing Problem

- Let $H_1, \ldots, H_N$ be a family of (hyper)graphs,
- $v(H_i) \leq n$ for each i, and

Edge disjoint copies – the packing problem

In a greater generality, one could state the following problem:

The Packing Problem

- Let $H_1, \ldots, H_N$ be a family of (hyper)graphs,
- $v(H_i) \leq n$ for each i, and
- $\sum e(H_i) \leq m$.

Edge disjoint copies – the packing problem

In a greater generality, one could state the following problem:

The Packing Problem

- Let $H_1, \ldots, H_N$ be a family of (hyper)graphs,
- $v(H_i) \leq n$ for each i, and
- $\sum e(H_i) \leq m$.

Question: Is it possible to find edge-disjoint copies of the H_is in a typical n-vertex (hyper)graph with $(1 + o(1))m$ edges?

Edge disjoint copies – online sprinkling

We describe the method introduced by F. and Vu which is widely applicable for finding 'many' edge-disjoint copies of H for various .
For example:

Edge disjoint copies – online sprinkling

We describe the method introduced by F. and Vu which is widely applicable for finding 'many' edge-disjoint copies of H for various .
For example:

- All the H_is are perfect matchings in a k-uniform hypergraph (F. and Vu '16).

Edge disjoint copies – online sprinkling

We describe the method introduced by F. and Vu which is widely applicable for finding 'many' edge-disjoint copies of H for various . For example:

- All the H_is are perfect matchings in a k-uniform hypergraph (F. and Vu '16).
- All the H_is are loose Hamilton cycles in a k-uniform hypergraph (F., Luh, Montealegre and Nguyen '16).

Edge disjoint copies – online sprinkling

We describe the method introduced by F. and Vu which is widely applicable for finding 'many' edge-disjoint copies of H for various .
For example:

- All the H_is are perfect matchings in a k-uniform hypergraph (F. and Vu '16).
- All the H_is are loose Hamilton cycles in a k-uniform hypergraph (F., Luh, Montealegre and Nguyen '16).
- Each H_i is a Hamilton cycle oriented arbitrarily (F. and Long '17).

Edge disjoint copies – online sprinkling

We describe the method introduced by F. and Vu which is widely applicable for finding 'many' edge-disjoint copies of H for various . For example:

- All the H_is are perfect matchings in a k-uniform hypergraph (F. and Vu '16).
- All the H_is are loose Hamilton cycles in a k-uniform hypergraph (F., Luh, Montealegre and Nguyen '16).
- Each H_i is a Hamilton cycle oriented arbitrarily (F. and Long '17).
- Each H_i is a rainbow Hamilton cycle (F. and Mousset '17).

Edge disjoint copies – online sprinkling

We describe the method introduced by F. and Vu which is widely applicable for finding 'many' edge-disjoint copies of H for various .
For example:

- All the H_is are perfect matchings in a k-uniform hypergraph (F. and Vu '16).
- All the H_is are loose Hamilton cycles in a k-uniform hypergraph (F., Luh, Montealegre and Nguyen '16).
- Each H_i is a Hamilton cycle oriented arbitrarily (F. and Long '17).
- Each H_i is a rainbow Hamilton cycle (F. and Mousset '17).
- All the H_is are trees (F. and Samotij '19).

Online sprinkling

Here we give a high level description of our online sprinkling method.

Online sprinkling

Here we give a high level description of our online sprinkling method.

Sprinkling: Let $p, p_1, \ldots, p_\ell \in [0,1]$ and suppose that

$$\prod_{i=1}^{\ell}(1-p_i) = 1-p.$$

Then, $H^{(k)}_{n,p}$ has the same distribution as $H = H_1 \cup \ldots \cup H_\ell$, where $H_i = H^{(k)}_{n,p_i}$ for each i.

Online sprinkling

Here we give a high level description of our online sprinkling method.

Sprinkling: Let $p, p_1, \ldots, p_\ell \in [0,1]$ and suppose that

$$\prod_{i=1}^{\ell}(1-p_i) = 1-p.$$

Then, $H_{n,p}^{(k)}$ has the same distribution as $H = H_1 \cup \ldots \cup H_\ell$, where $H_i = H_{n,p_i}^{(k)}$ for each i.

Online Sprinkling: A randomized algorithm s.t during its execution, a random structure is generated online and a target structure is constructed together step by step.

Online sprinkling

Here we give a high level description of our online sprinkling method.

Sprinkling: Let $p, p_1, \ldots, p_\ell \in [0,1]$ and suppose that

$$\prod_{i=1}^{\ell}(1-p_i) = 1-p.$$

Then, $H_{n,p}^{(k)}$ has the same distribution as $H = H_1 \cup \ldots \cup H_\ell$, where $H_i = H_{n,p_i}^{(k)}$ for each i.

Online Sprinkling: A randomized algorithm s.t during its execution, a random structure is generated online and a target structure is constructed together step by step.

Online sprinkling

How does it work?

Online sprinkling

How does it work?

- In each time step i of the algorithm, a subset $E_i \subseteq E(K_n)$ is chosen according to some distribution (which depends on the history).

Online sprinkling

How does it work?

- In each time step i of the algorithm, a subset $E_i \subseteq E(K_n)$ is chosen according to some distribution (which depends on the history).
- Then, edges in E_i are chosen (independently) with some probability p_i (which depends on the history). All the chosen edges will be part of the randomly generated structure.

Online sprinkling

How does it work?

- In each time step i of the algorithm, a subset $E_i \subseteq E(K_n)$ is chosen according to some distribution (which depends on the history).
- Then, edges in E_i are chosen (independently) with some probability p_i (which depends on the history). All the chosen edges will be part of the randomly generated structure.
- For each k-tuple $e \in \binom{[n]}{k}$, let

$$\omega(e) = 1 - \prod_{i : e \in E_i} (1 - p_i)$$

be the weight of e at the end of the algorithm.

Online sprinkling

How does it work?

- In each time step i of the algorithm, a subset $E_i \subseteq E(K_n)$ is chosen according to some distribution (which depends on the history).
- Then, edges in E_i are chosen (independently) with some probability p_i (which depends on the history). All the chosen edges will be part of the randomly generated structure.
- For each k-tuple $e \in \binom{[n]}{k}$, let

$$\omega(e) = 1 - \prod_{i : e \in E_i} (1 - p_i)$$

  be the weight of e at the end of the algorithm.
- Clearly, if $\omega(e) \leq p$ for each k-tuple e, then the resulting structure can be coupled with a subgraph of $H \sim \mathcal{H}^k(n, p)$.

Resilience

Recall the definition of resilience.

Resilience

Recall the definition of resilience. Sudakov and Vu 2008 proved that for $p \gg \log n/n$ a typical $G_{n,p}$ is $(1/2 + o(1))$-resilient under the property 'having a perfect matching'.

Resilience

Recall the definition of resilience. Sudakov and Vu 2008 proved that for $p \gg \log n/n$ a typical $G_{n,p}$ is $(1/2 + o(1))$-resilient under the property 'having a perfect matching'.

Tight results for Hamiltonicity (in both random undirected and directed models) are also currently known (Lee and Sudakov, Montgomery, respectively).

Resilience

Recall the definition of resilience. Sudakov and Vu 2008 proved that for $p \gg \log n/n$ a typical $G_{n,p}$ is $(1/2 + o(1))$-resilient under the property 'having a perfect matching'.

Tight results for Hamiltonicity (in both random undirected and directed models) are also currently known (Lee and Sudakov, Montgomery, respectively). .

For hypergraphs, the main problem is that a Dirac-type result is unkowon in the dense case (there is a neat connection between this problem and some famous probabilistic conjecture by Feige).

Resilience

Recall the definition of resilience. Sudakov and Vu 2008 proved that for $p \gg \log n/n$ a typical $G_{n,p}$ is $(1/2 + o(1))$-resilient under the property 'having a perfect matching'.

Tight results for Hamiltonicity (in both random undirected and directed models) are also currently known (Lee and Sudakov, Montgomery, respectively). .

For hypergraphs, the main problem is that a Dirac-type result is unkowon in the dense case (there is a neat connection between this problem and some famous probabilistic conjecture by Feige).

BUT very recently we proved

Theorem (F. and Kwan 2020+)

For $p \geq n^{-k/2+o(1)}$, a typical $H_{n,p}^{(k)}$ is $\mu_d(k)$-resilient, where $\mu_d(k)$ is the optimal (and yet, unknown) ratio $\delta_d(k,n)/\binom{n}{k}$ that enforces a perfect matching.

Maker-Breaker games

An $(\mathcal{F}, (a, b))$-game is define as follows:

- V board and $\mathcal{F} \subseteq 2^V$ winning sets.
- Two players, Maker and Breaker, alternating turns in occupying a and b elements, respectively.
- Maker wins if and only if she claims all the elements of a winning set.

Maker-Breaker games

An $(\mathcal{F}, (a, b))$-game is define as follows:

- V board and $\mathcal{F} \subseteq 2^V$ winning sets.
- Two players, Maker and Breaker, alternating turns in occupying a and b elements, respectively.
- Maker wins if and only if she claims all the elements of a winning set.

Well studied and have a long history.

Maker-Breaker games

An $(\mathcal{F}, (a, b))$-game is define as follows:

- V board and $\mathcal{F} \subseteq 2^V$ winning sets.
- Two players, Maker and Breaker, alternating turns in occupying a and b elements, respectively.
- Maker wins if and only if she claims all the elements of a winning set.

Well studied and have a long history. One neat connection to random graphs:

Theorem (F.,Krivelevich and Naves 2015)

Let $p >> \log n/n^{k-1}$. Then, in the $(1, \varepsilon/p)$-game played on $E(K_n^{(k)})$, maker has a randomized strategy to build a subgraph $H \subseteq G = H_{n,p}^{(k)}$ for which $d_H(v) \geq (1 - f(\varepsilon))d_G(v)$ for all v.

Maker-Breaker games

An $(\mathcal{F}, (a, b))$-game is define as follows:

- V board and $\mathcal{F} \subseteq 2^V$ winning sets.
- Two players, Maker and Breaker, alternating turns in occupying a and b elements, respectively.
- Maker wins if and only if she claims all the elements of a winning set.

Well studied and have a long history. One neat connection to random graphs:

Theorem (F.,Krivelevich and Naves 2015)

Let $p >> \log n/n^{k-1}$. Then, in the $(1, \varepsilon/p)$-game played on $E(K_n^{(k)})$, maker has a randomized strategy to build a subgraph $H \subseteq G = H_{n,p}^{(k)}$ for which $d_H(v) \geq (1 - f(\varepsilon))d_G(v)$ for all v.

That is, if we have a (weak) local resilience result for $H_{n,p}^{(k)}$, it can be immediately translated into a Maker-Breaker result!

End of time :-)

If you are interested at hearing more, don't hesitate to contact me!
asaff@uci.edu

Thank you for listening!

Questions?