# Hypergraph 2-coloring

## The best known algorithm and its limitations

Grzegorz Ryn

April 11, 2025

# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

## Hypergraph

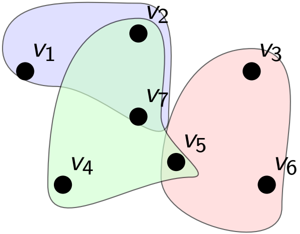
A hypergraph $\mathcal{H} = (V, E)$ is a pair consisting of a set of vertices $V$ and a set of (hyper)edges $E \subseteq \mathcal{P}(V)$.

# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

## Hypergraph

A hypergraph $\mathcal{H} = (V, E)$ is a pair consisting of a set of vertices $V$ and a set of (hyper)edges $E \subseteq \mathcal{P}(V)$.



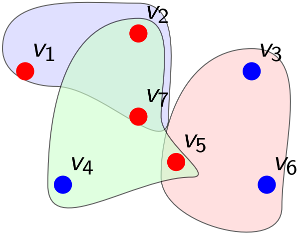
# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

## Hypergraph

A hypergraph $\mathcal{H} = (V, E)$ is a pair consisting of a set of vertices $V$ and a set of (hyper)edges $E \subseteq \mathcal{P}(V)$.

- A (hyper)graph coloring is a functions $c : V \to \mathbb{N}$



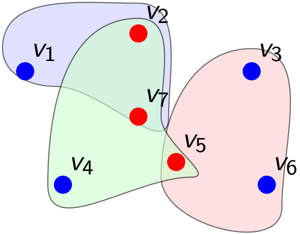
# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

## Hypergraph

A hypergraph $\mathcal{H} = (V, E)$ is a pair consisting of a set of vertices $V$ and a set of (hyper)edges $E \subseteq \mathcal{P}(V)$.

- A (hyper)graph coloring is a functions $c : V \to \mathbb{N}$
- coloring is proper if no edge is monochromatic



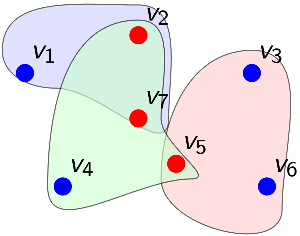
# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

## Hypergraph

A hypergraph $\mathcal{H} = (V, E)$ is a pair consisting of a set of vertices $V$ and a set of (hyper)edges $E \subseteq \mathcal{P}(V)$.

- A (hyper)graph coloring is a functions $c : V \to \mathbb{N}$
- coloring is proper if no edge is monochromatic
- Function $c$ is a $k$-coloring if it uses at most $k$ different colors



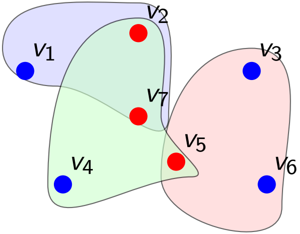
# Graphs and hypergraphs

An undirected *graph* is a pair $(V, E)$ consisting of a set of vertices $V$ and edges $E \subseteq V^2$. This yields a natural generalization.

## Hypergraph

A hypergraph $\mathcal{H} = (V, E)$ is a pair consisting of a set of vertices $V$ and a set of (hyper)edges $E \subseteq \mathcal{P}(V)$.

- A (hyper)graph coloring is a functions $c : V \to \mathbb{N}$
- coloring is proper if no edge is monochromatic
- Function $c$ is a $k$-coloring if it uses at most $k$ different colors




A hypergraph with all edges of size $k$ is called $k$-uniform or $k$-graph.

# Random hypergraphs

The most widely known random graph generation model is Erdős–Rényi model or $G(n, p)$.

## Erdős–Rényi model

In $G(n, p)$ model, we obtain a random graph on $n$ vertices by independently adding each of $\binom{n}{2}$ possible edges with probability $p$.

# Random hypergraphs

The most widely known random graph generation model is Erdős–Rényi model or $G(n, p)$.

## Erdős–Rényi model

In $G(n, p)$ model, we obtain a random graph on $n$ vertices by independently adding each of $\binom{n}{2}$ possible edges with probability $p$.

We can naturally generalize it to $k$-uniform hypergraphs.

## $H_k(n, p)$ model

In $H_k(n, p)$ model, we obtain a random $k$-uniform hypergraph on $n$ vertices by independently including each of $\binom{n}{k}$ possible edges with probability $p$.

# Random hypergraphs

The most widely known random graph generation model is Erdős–Rényi model or $G(n, p)$.

## Erdős–Rényi model

In $G(n, p)$ model, we obtain a random graph on $n$ vertices by independently adding each of $\binom{n}{2}$ possible edges with probability $p$.

We can naturally generalize it to $k$-uniform hypergraphs.

## $H_k(n, p)$ model

In $H_k(n, p)$ model, we obtain a random $k$-uniform hypergraph on $n$ vertices by independently including each of $\binom{n}{k}$ possible edges with probability $p$.

We can generalize it further to non-uniform random hypergraphs by considering each size of (allowed) edge independently.

# 2-colorability

## (hyper)graph 2-colorability

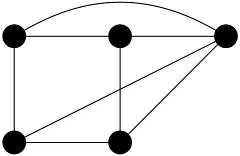
Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.

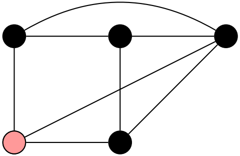


# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.

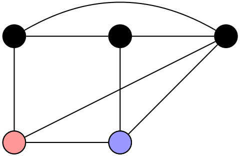


# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.

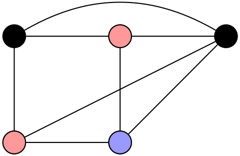


# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.

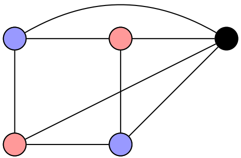


# 2-colorability

**(hyper)graph 2-colorability**

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.

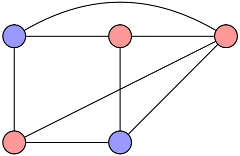


# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.

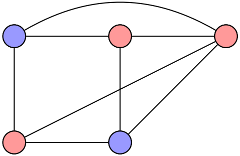


# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.



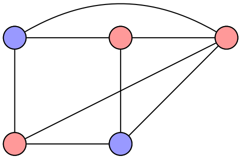
It's easy to check, that graph is 2-colorable iff it doesn't contain an odd cycle.

# 2-colorability

## (hyper)graph 2-colorability

Given a (hyper)graph $\mathcal{H}$, determine if there exists its proper 2-coloring - i.e. $\mathcal{H}$ is 2-colorable.

Let's first have a look at the "regular" version of 2-colorability. Since we want to find out if given graph has a valid 2-coloring and each edge consists of exactly two vertices, we can just greedily assign alternative colors and check for contradiction.



It's easy to check, that graph is 2-colorable iff it doesn't contain an odd cycle.

## $k$-uniform graph 2-colorability

In 1973, L. Lovász showed, that for $k \geq 3$, 2-colorability of $k$-uniform Hypergraph is NP-complete.

# Setting

We are interested in an "assymptotic" setting. Let $k$ be fixed and $n$ arbitrarily large. Obviously hypergraphs drawn from $H_k(n, 0)$ are 2-colorable with high probability but drawn from $H_k(n, 1)$ aren't.

# Setting

We are interested in an "assymptotic" setting. Let $k$ be fixed and $n$ arbitrarily large. Obviously hypergraphs drawn from $H_k(n, 0)$ are 2-colorable with high probability but drawn from $H_k(n, 1)$ aren't.

## Our goal

What is the maximal value of $p = p(n)$ s.t. hypergraph drawn from $H_k(n, p)$ is (algorithmically) 2-colorable with high probability?

# Setting

We are interested in an "assymptotic" setting. Let $k$ be fixed and $n$ arbitrarily large. Obviously hypergraphs drawn from $H_k(n, 0)$ are 2-colorable with high probability but drawn from $H_k(n, 1)$ aren't.

## Our goal

What is the maximal value of $p = p(n)$ s.t. hypergraph drawn from $H_k(n, p)$ is (algorithmically) 2-colorable with high probability?

As we might find out, working explicitly with the value of $p$ isn't convenient. Instead we will parametrize our random graph model by the expected number of edges.

# Boundaries

First, we may recall the upper bound. It is based on an idea of Erdős, but in this setting was shown by Alon and Spencer that:

## Upper bound, Erdős 1964

Random hypergraph on $n$ vertices with $\frac{\ln 2}{2} \cdot 2^k \cdot n$ edges on average is with high probability not-2-colorable.

# Boundaries

First, we may recall the upper bound. It is based on an idea of Erdős, but in this setting was shown by Alon and Spencer that:

**Upper bound, Erdős 1964**

Random hypergraph on $n$ vertices with $\frac{\ln 2}{2} \cdot 2^k \cdot n$ edges on average is with high probability not-2-colorable.

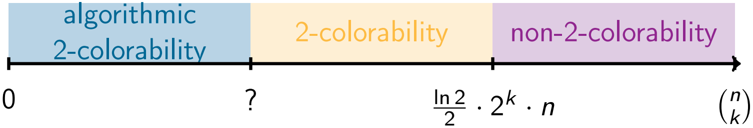
The search for lower bound has been more convoluted. There have been a number of improvments, with the last big improvments in 2001 by Achlioptas and Moore.

**Lower bound, Achlioptas and Moore 2001**

Random hypergraph on $n$ vertices with $\left(\frac{\ln 2}{2} \cdot \left(2^k - 1\right) - \varepsilon\right) \cdot n$ edges on average is 2-colorable with high probability.

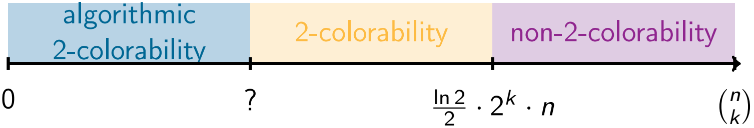
# Bird's-eye view

Knowing the threshold at which random hypergraphs stop being 2-colorable, we may ask 'How dense graph can we color with *fast* algorithm?



# Bird's-eye view

Knowing the threshold at which random hypergraphs stop being 2-colorable, we may ask 'How dense graph can we color with *fast* algorithm?




On the sidenote: on the graphic above the intervals corresponding to *2-colorability* and *non-2-colorability* are drawn just after each other. It is not only a simplification, but a reflection of how rapidly our model changes its behaviour. In fact if we look at the values of $p$ for which random graphs are neither 2-colorable or non-2-colorable with high probability, the length of such interval is completely negligible.

# AKKT algorithm

Having build up some context, we may introduce the best algorithm for hypergraph 2-coloring. It was introduced in 1999 by Achlioptas, Kim, Krivelevich and Tetali, hence we will refer to it as **AKKT**.

# AKKT algorithm

Having build up some context, we may introduce the best algorithm for hypergraph 2-coloring. It was introduced in 1999 by Achlioptas, Kim, Krivelevich and Tetali, hence we will refer to it as **AKKT**.

### AKKT 2-colorability threshold

The AKKT algorithm suceeds (produces a proper 2-coloring) for random hypergraphs on $n$ vertices with at most $\lambda \cdot \frac{2^k}{k} \cdot n$ edges on a average.

# AKKT algorithm

Having build up some context, we may introduce the best algorithm for hypergraph 2-coloring. It was introduced in 1999 by Achlioptas, Kim, Krivelevich and Tetali, hence we will refer to it as **AKKT**.

## AKKT 2-colorability threshold

The AKKT algorithm suceeds (produces a proper 2-coloring) for random hypergraphs on $n$ vertices with at most $\lambda \cdot \frac{2^k}{k} \cdot n$ edges on a average.

Moreover, at the time of its first appearance, this threshold was not only the best-known threshold for algorithmic 2-colroability, but for 2-colorability in general.

# AKKT algorithm

## AKKT algorithm

- Our algorithm will perform $n/2$ phases. In each of them it will pick two vertices and color one of them *RED*and the other *BLUE*.

# AKKT algorithm

## AKKT algorithm

- Our algorithm will perform $n/2$ phases. In each of them it will pick two vertices and color one of them *RED*and the other *BLUE*.
- If there exists an edge with only 2 or 3 uncolored vertices, with the rest being monochromatic, pick any such edge and pick two of its uncolored vertices.

# AKKT algorithm

## AKKT algorithm

- Our algorithm will perform $n/2$ phases. In each of them it will pick two vertices and color one of them *RED*and the other *BLUE*.
- If there exists an edge with only 2 or 3 uncolored vertices, with the rest being monochromatic, pick any such edge and pick two of its uncolored vertices.
- Otherwise, pick any two vertices.

# AKKT algorithm

## AKKT algorithm

- Our algorithm will perform $n/2$ phases. In each of them it will pick two vertices and color one of them *RED*and the other *BLUE*.
- If there exists an edge with only 2 or 3 uncolored vertices, with the rest being monochromatic, pick any such edge and pick two of its uncolored vertices.
- Otherwise, pick any two vertices.

Let's call a $k$-edge with $l$ uncolored vertice and $(k-l)$ monochromatic vertices an $l$-tail. It's quite intuitive, that for our algorithm to fail, at some point in time there needs to be **a lot** of 2 and 3 tails.

# AKKT algorithm

## AKKT algorithm

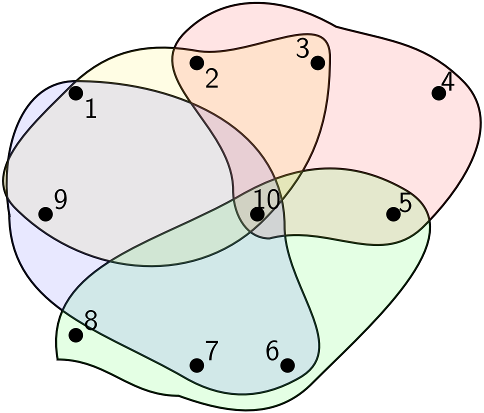
- Our algorithm will perform $n/2$ phases. In each of them it will pick two vertices and color one of them *RED*and the other *BLUE*.
- If there exists an edge with only 2 or 3 uncolored vertices, with the rest being monochromatic, pick any such edge and pick two of its uncolored vertices.
- Otherwise, pick any two vertices.

Let's call a $k$-edge with $l$ uncolored vertice and $(k - l)$ monochromatic vertices an $l$-tail. It's quite intuitive, that for our algorithm to fail, at some point in time there needs to be **a lot** of 2 and 3 tails.

We can think of the currently existing 2 and 3 tails as an auxilary hypergraph (on uncolored vertices) with edges of size 2 and 3.

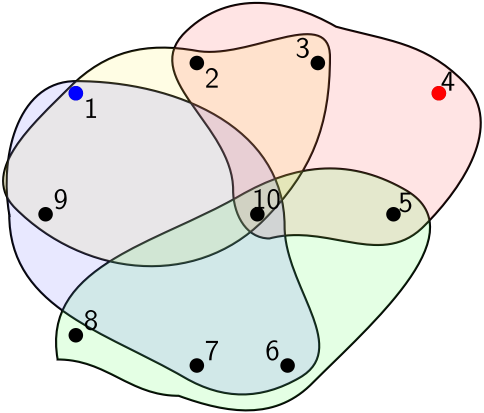
# AKKT example

## hypergraph to be colored



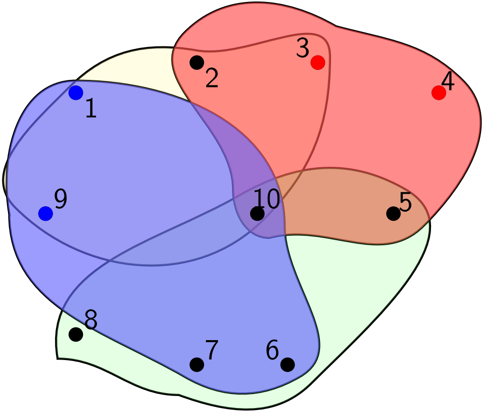
# AKKT example

**hypergraph to be colored**

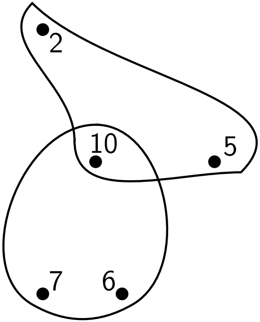


# AKKT example

hypergraph to be colored

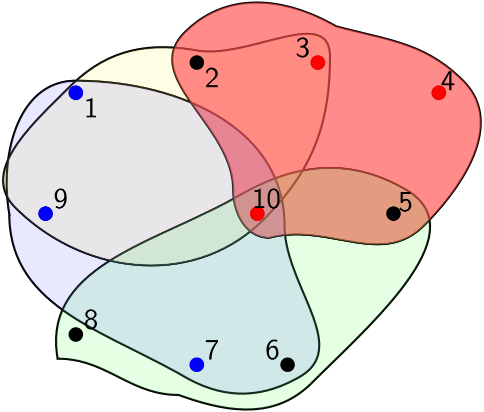



auxilary hypergraph

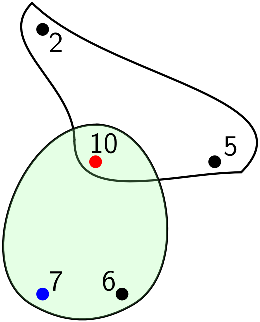


# AKKT example

## hypergraph to be colored

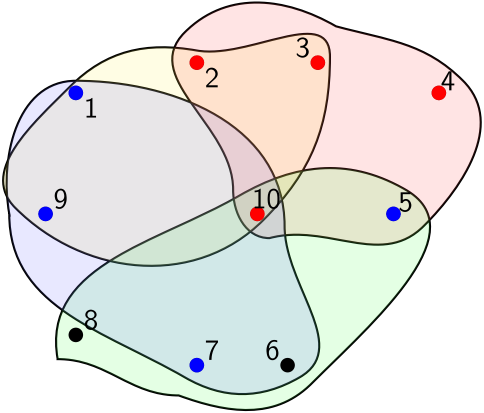



## auxilary hypergraph

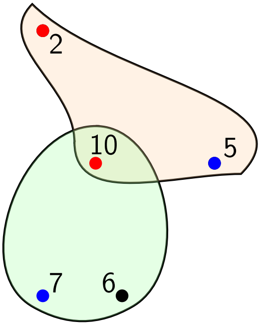


# AKKT example

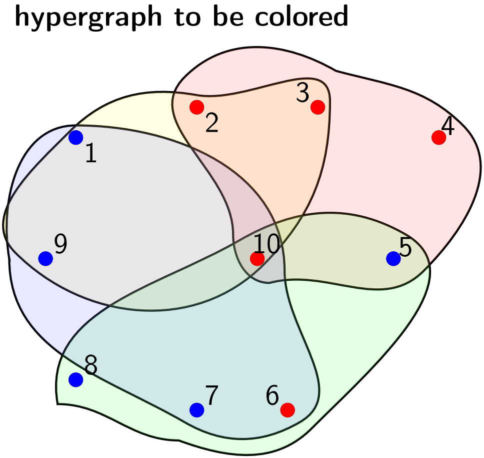
## hypergraph to be colored

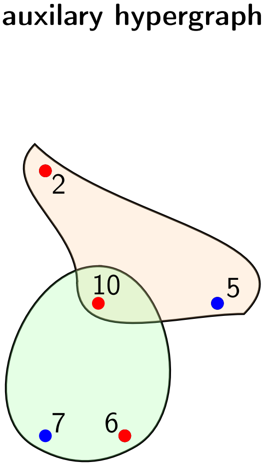



## auxilary hypergraph



# AKKT example



# Auxilary hypergraph

As we have seen, the *interesting* part of the algorithm only takes place, when new edges appear and are colored in the auxilary graph.

# Auxilary hypergraph

As we have seen, the *interesting* part of the algorithm only takes place, when new edges appear and are colored in the auxilary graph.

We can look at this algorithm from a different perspective. Let's forget about underlaying hypergraph and focus on coloring newly appearing edges in its auxilary graph. It turns out that the following problem is equivalent:

# Auxilary hypergraph

As we have seen, the *interesting* part of the algorithm only takes place, when new edges appear and are colored in the auxilary graph.

We can look at this algorithm from a different perspective. Let's forget about underlaying hypergraph and focus on coloring newly appearing edges in its auxilary graph. It turns out that the following problem is equivalent:

Let's consider a hypergraph on uncolored vertices only. In each of $n/2$ round of the algorithm, two vertices are collored (differently) and a random number of new 3-edges appear in a narrowed graph.

# Distribution of new edges

## Number of newly appeared edges

It turns out, that in each step of the algorithm, the number of new edges is a Poisson variable.

# Distribution of new edges

## Number of newly appeared edges

It turns out, that in each step of the algorithm, the number of new edges is a Poisson variable.

We may ask the question "How many new edges do we expect to see in each step?"

# Distribution of new edges

## Number of newly appeared edges

It turns out, that in each step of the algorithm, the number of new edges is a Poisson variable.

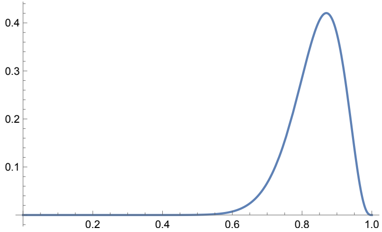
We may ask the question "How many new edges do we expect to see in each step?" It turns out, that almost nothing ever happens.

# Distribution of new edges

## Number of newly appeared edges

It turns out, that in each step of the algorithm, the number of new edges is a Poisson variable.

We may ask the question "How many new edges do we expect to see in each step?" It turns out, that almost nothing ever happens.



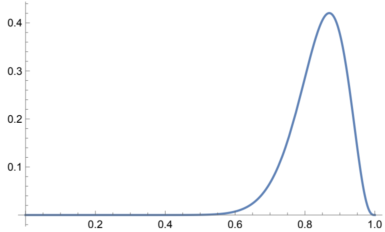

$k = 20$

# Distribution of new edges

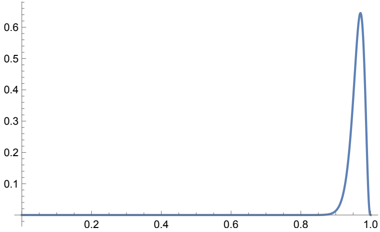
## Number of newly appeared edges

It turns out, that in each step of the algorithm, the number of new edges is a Poisson variable.

We may ask the question "How many new edges do we expect to see in each step?" It turns out, that almost nothing ever happens.




$k = 20$



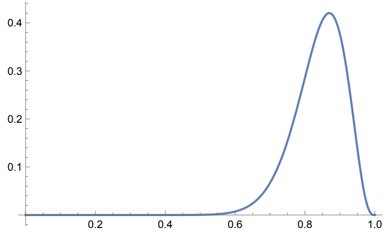

$k = 100$

# Distribution of new edges

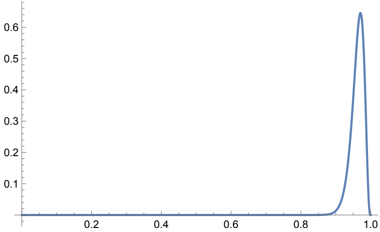
## Number of newly appeared edges

It turns out, that in each step of the algorithm, the number of new edges is a Poisson variable.

We may ask the question "How many new edges do we expect to see in each step?" It turns out, that almost nothing ever happens.




$k = 20$




$k = 100$

It can be shown, that we can w.h.p. 2-color our hypergraph if at any point, the expected number of new edges is less than 1.

# non-uniform case

Having gained some intuition on uniform case, we may investigate non-uniform hypergraphs.

# non-uniform case

Having gained some intuition on uniform case, we may investigate non-uniform hypergraphs.

Since, for different values of $k$, the peaks in the distribution of 3-tails appear in different phases of the execution, we might want to try to 2-color "larger" hypergraphs by allowing different edge sizes.

# non-uniform case

Having gained some intuition on uniform case, we may investigate non-uniform hypergraphs.

Since, for different values of $k$, the peaks in the distribution of 3-tails appear in different phases of the execution, we might want to try to 2-color "larger" hypergraphs by allowing different edge sizes.

**AKKT, uniform case**

AKKT suceeds (w.h.p) for graphs with at most $c \cdot \frac{2^k}{k} \cdot n$ edges on average.

# non-uniform case

Having gained some intuition on uniform case, we may investigate non-uniform hypergraphs.

Since, for different values of $k$, the peaks in the distribution of 3-tails appear in different phases of the execution, we might want to try to 2-color "larger" hypergraphs by allowing different edge sizes.

**AKKT, uniform case**

AKKT suceeds (w.h.p) for graphs with at most $c \cdot \frac{2^k}{k} \cdot n$ edges on average. In particular it fails for $c \approx 1$.

# non-uniform case

Having gained some intuition on uniform case, we may investigate non-uniform hypergraphs.

Since, for different values of $k$, the peaks in the distribution of 3-tails appear in different phases of the execution, we might want to try to 2-color "larger" hypergraphs by allowing different edge sizes.

**AKKT, uniform case**

AKKT suceeds (w.h.p) for graphs with at most $c \cdot \frac{2^k}{k} \cdot n$ edges on average. In particular it fails for $c \approx 1$.

Let's work with the non-uniform model allowing for edges of sizes $k_1, k_2, \ldots, k_l$ with expectancy $\lambda_1 \cdot \frac{2^{k_1}}{k_1} \cdot n, \lambda_2 \cdot \frac{2^{k_2}}{k_2} \cdot n, \ldots$.

# non-uniform case

Having gained some intuition on uniform case, we may investigate non-uniform hypergraphs.

Since, for different values of $k$, the peaks in the distribution of 3-tails appear in different phases of the execution, we might want to try to 2-color "larger" hypergraphs by allowing different edge sizes.

**AKKT, uniform case**

AKKT suceeds (w.h.p) for graphs with at most $c \cdot \frac{2^k}{k} \cdot n$ edges on average. In particular it fails for $c \approx 1$.

Let's work with the non-uniform model allowing for edges of sizes $k_1, k_2, \ldots, k_l$ with expectancy $\lambda_1 \cdot \frac{2^{k_1}}{k_1} \cdot n, \lambda_2 \cdot \frac{2^{k_2}}{k_2} \cdot n, \ldots$.

**AKKT, non-uniform case**

For any $c > 0$, there exists a corresponding random hypergraph model (a sequence of $(k_i, \lambda_i)$), s.t: $\sum \lambda_i \geq c$

# non-uniform case - upper bound

Let's recall the upper bound (non-constructive) for 2-colorability:

**Upper bound, Erdős 1964**

Random hypergraph on $n$ vertices with $\frac{\ln 2}{2} \cdot 2^k \cdot n$ edges on average is with high probability not-2-colorable.

# non-uniform case - upper bound

Let's recall the upper bound (non-constructive) for 2-colorability:

**Upper bound, Erdős 1964**

Random hypergraph on $n$ vertices with $\frac{\ln 2}{2} \cdot 2^k \cdot n$ edges on average is with high probability not-2-colorable.

Once again, woriking with analogous non-uniform model allowing for edges of sizes $k_1, k_2, \ldots, k_l$ with the corresponding expectancy: $\lambda_1 \cdot 2^{k_1} \cdot n, \lambda_2 \cdot 2^{k_2} \cdot n, \ldots$:

# non-uniform case - upper bound

Let's recall the upper bound (non-constructive) for 2-colorability:

**Upper bound, Erdős 1964**

Random hypergraph on $n$ vertices with $\frac{\ln 2}{2} \cdot 2^k \cdot n$ edges on average is with high probability not-2-colorable.

Once again, woriking with analogous non-uniform model allowing for edges of sizes $k_1, k_2, \ldots, k_l$ with the corresponding expectancy: $\lambda_1 \cdot 2^{k_1} \cdot n, \lambda_2 \cdot 2^{k_2} \cdot n, \ldots$:

**non-uniform upper bound**

Random non-uniform hypergraph drawn from a model described above is with high probability non-2-colorable if $\sum \lambda_i \geq \frac{\ln 2}{2}$.

# interesting contrast

We can observe, that the effect of introducing non-uniformity is different for algorithmic and non-constructive approach.

# interesting contrast

We can observe, that the effect of introducing non-uniformity is different for algorithmic and non-constructive approach.

We can think of $2^k \cdot n$ and $\frac{2^k}{k} \cdot n$ as of an assymptotic order of the number of edges *critical* in critical hypergraphs - for 2-colorability and algorithmic 2-colorability respectively.

# interesting contrast

We can observe, that the effect of introducing non-uniformity is different for algorithmic and non-constructive approach.

We can think of $2^k \cdot n$ and $\frac{2^k}{k} \cdot n$ as of an assymptotic order of the number of edges *critical* in critical hypergraphs - for 2-colorability and algorithmic 2-colorability respectively.

## Algorithmic vs. non-constructive upper bound

| | algorithmic* | non-constructive |
|---|---|---|
| **uniform** | $\approx 1/10$ | $\frac{\ln 2}{2}$ |
| **non-uniform** | unbounded | $\frac{\ln 2}{2}$ |

# interesting contrast

We can observe, that the effect of introducing non-uniformity is different for algorithmic and non-constructive approach.

We can think of $2^k \cdot n$ and $\frac{2^k}{k} \cdot n$ as of an assymptotic order of the number of edges *critical* in critical hypergraphs - for 2-colorability and algorithmic 2-colorability respectively.

## Algorithmic vs. non-constructive upper bound

| | **algorithmic*** | **non-constructive** |
|---|---|---|
| **uniform** | $\approx 1/10$ | $\frac{\ln 2}{2}$ |
| **non-uniform** | unbounded | $\frac{\ln 2}{2}$ |

It shows, that while 2-colorability threshold translates 1:1 between uniform and non-uniform case. Moreover hypergraphs are in some sense invariant to the type of an edge, while there's some space left in algorithmic coloring.

# Algorithmic barrier

To conclude, we may ask the question - can we color denser hypergraphs?

# Algorithmic barrier

To conclude, we may ask the question - can we color denser hypergraphs?

We don't know, but have a strong reason to believe that $\frac{2^k}{k} \cdot n$ is indeed a boundary assymptotic order for algorithmic 2-colorability.

# Algorithmic barrier

To conclude, we may ask the question - can we color denser hypergraphs?

We don't know, but have a strong reason to believe that $\frac{2^k}{k} \cdot n$ is indeed a boundary assymptotic order for algorithmic 2-colorability.

We can consider the space of all $2^n$ 2-colorings of $n$-vertex hypergraph. In 2008, Coja-Oghlan published a paper in which he analyses how the geometry of this space changes with increasing $p$.

# Algorithmic barrier

To conclude, we may ask the question - can we color denser hypergraphs?

We don't know, but have a strong reason to believe that $\frac{2^k}{k} \cdot n$ is indeed a boundary assymptotic order for algorithmic 2-colorability.

We can consider the space of all $2^n$ 2-colorings of $n$-vertex hypergraph. In 2008, Coja-Oghlan published a paper in which he analyses how the geometry of this space changes with increasing $p$.

It happens, that for $p$ up to around $\frac{2^k}{k} \cdot n$ the space of proper colorings is connected in the sense of Hamming distance. Beyond this threshold, this space becomes *shattered*. From the connected space, it breaks into exponentionally many small pieces, which are very far apart from each other - linearly in terms of vertices.