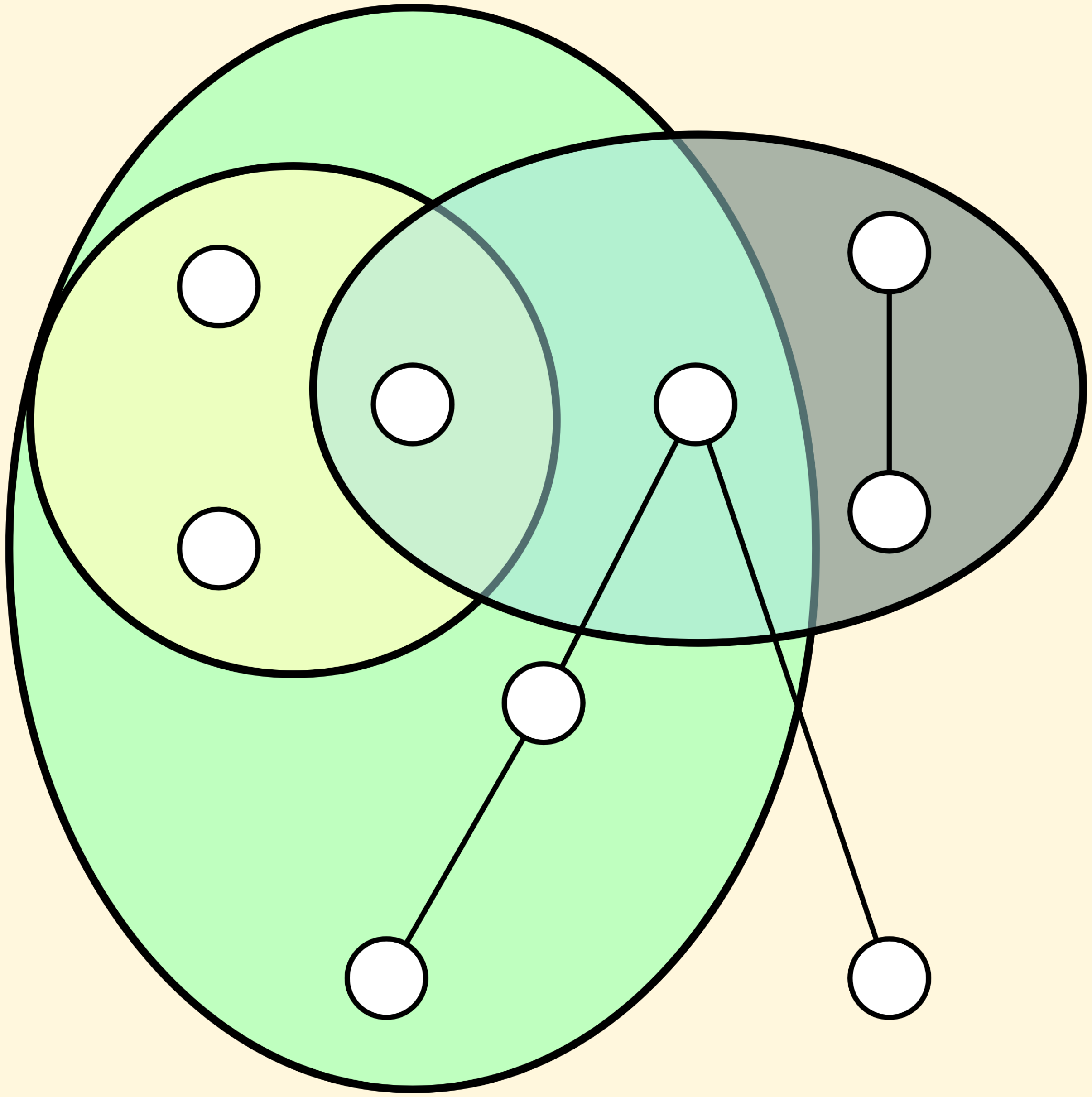
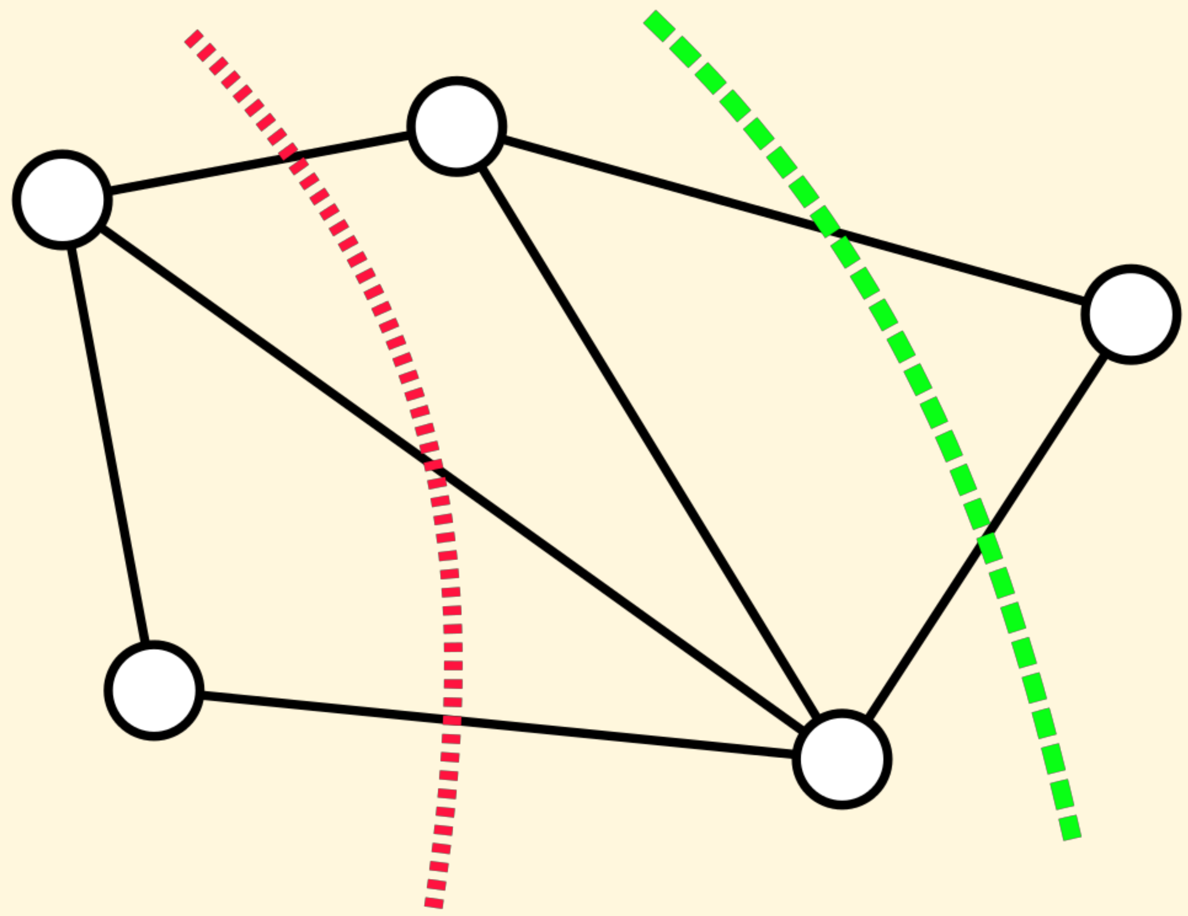


Minimum Cut and Minimum *k*-Cut in Hypergraphs via Branching Contractions

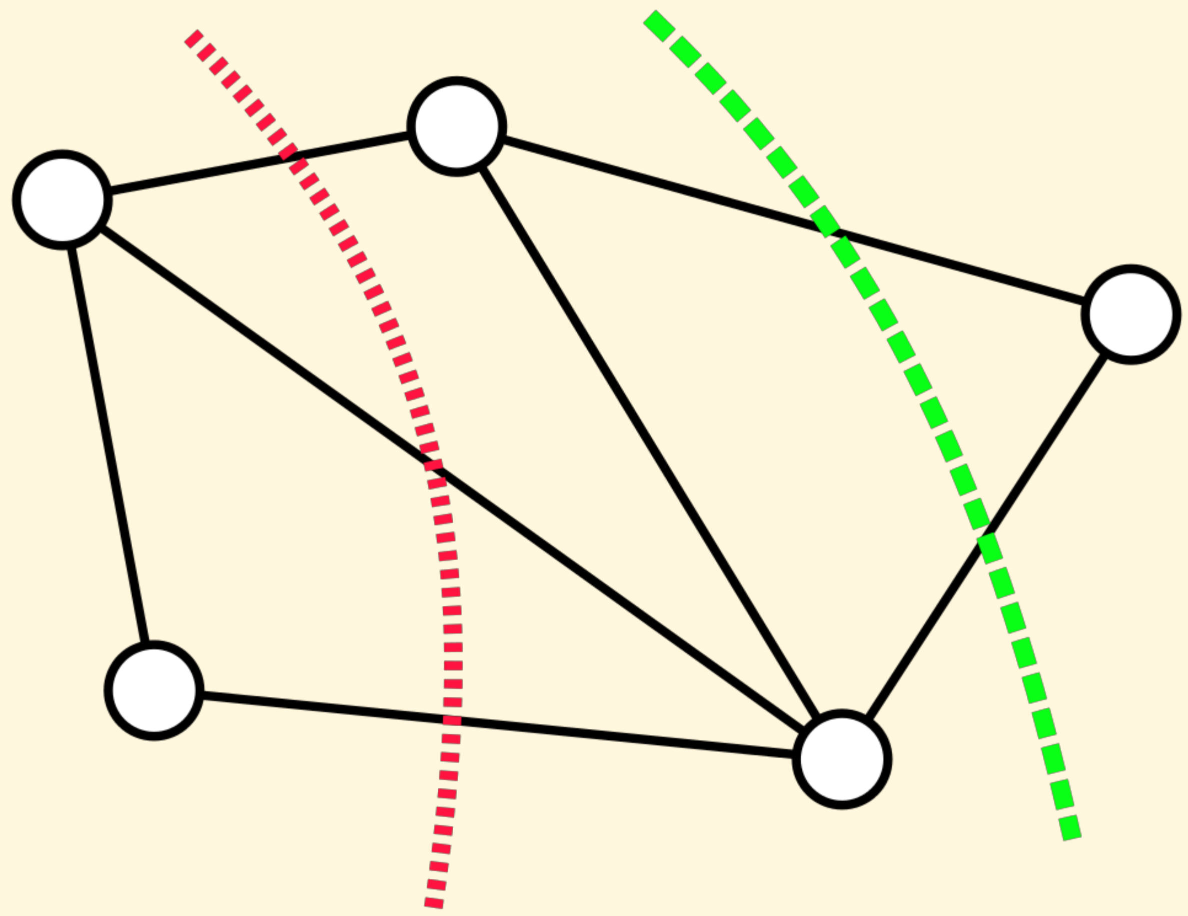
Fred Zhang (Harvard)

A joint work with Kyle Fox (UT-Dallas) and Debmalya Panigrahi (Duke)

Graphs and Cuts

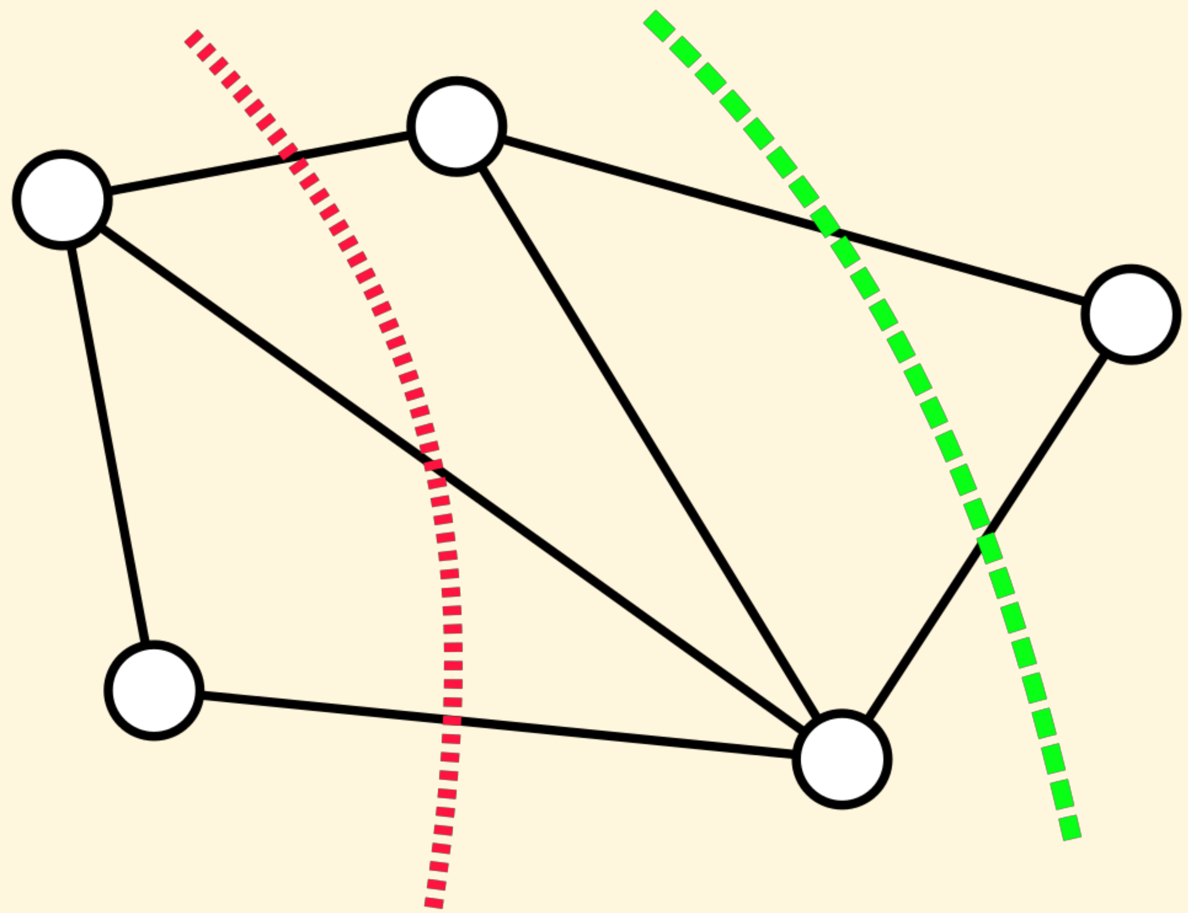


Graphs and Cuts



Min-Cut: minimal set of edges whose removal disconnects the graph

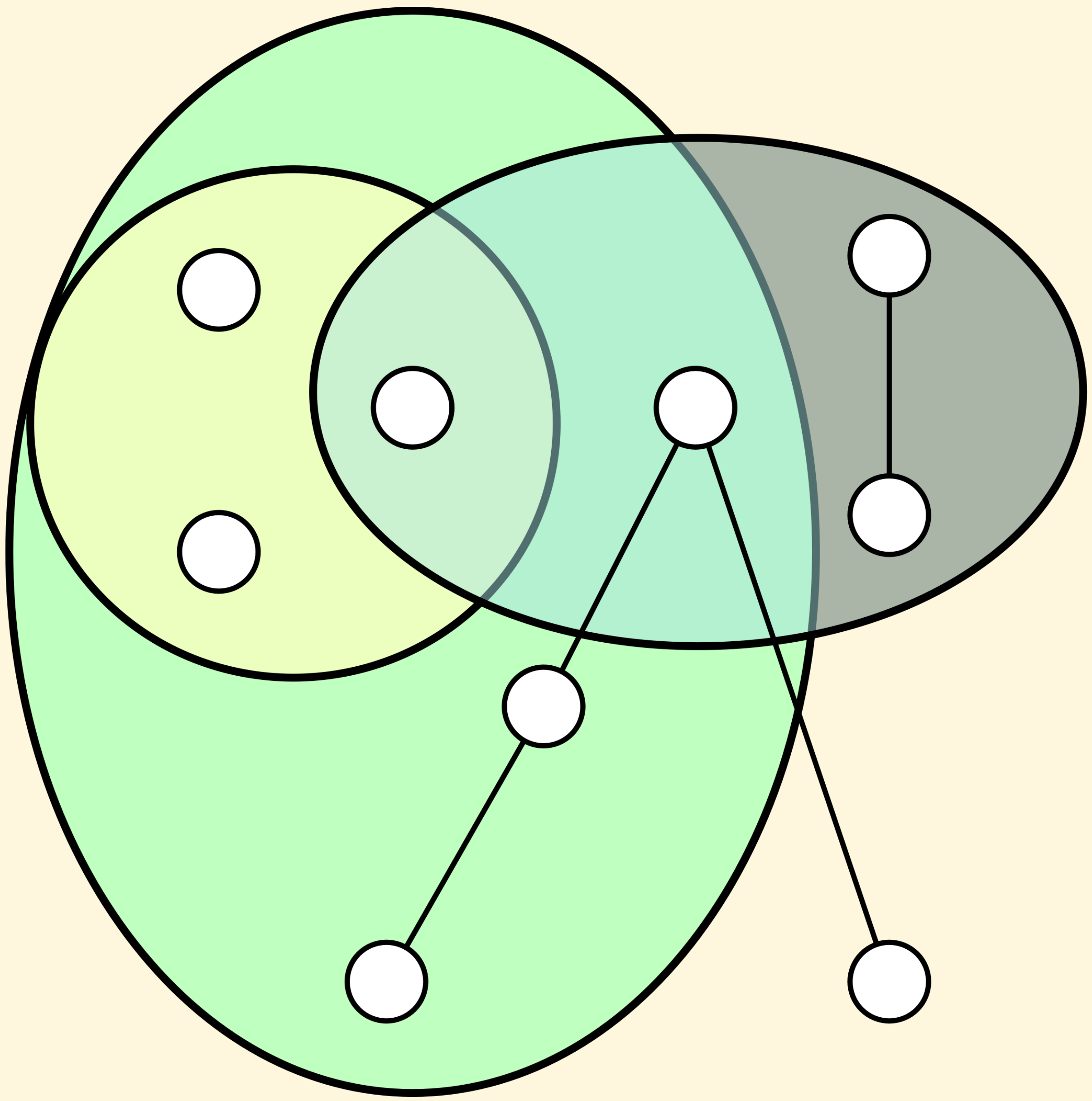
Graphs and Cuts



Min-Cut: minimal set of edges whose removal disconnects the graph

Min-*k*-Cut: minimal set of edges whose removal disconnects the graph into *k* components

Hypergraphs and Cuts



Goal: Compute min-k-cut of a (weighted) hypergraph

Previous works (weighted setting)

Given a hypergraph with n vertices, m edges, and $p = \Sigma |e|$,

Previous works (weighted setting)

Given a hypergraph with n vertices, m edges, and $p = \Sigma |e|$,

- **Min-Cut** in time $O(pn+n^2\lambda)$, where λ is the min-cut size
  [Klimmek-Wagner, '96] [Mak-Wong, '00]

Previous works (weighted setting)

Given a hypergraph with n vertices, m edges, and $p = \Sigma |e|$,

- **Min-Cut** in time $O(pn+n^2\lambda)$, where λ is the min-cut size
  [Klimmek-Wagner, '96] [Mak-Wong, '00]

- **Min-*k*-Cut** in time $O(pn^{2k-1})$ for constant k
  [Chandrasekaran-Xu-Yu, '18]

Previous works (weighted setting)

Given a hypergraph with n vertices, m edges, and $p = \Sigma |e|$,

- **Min-Cut** in time $O(pn+n^2\lambda)$, where λ is the min-cut size
  [Klimmek-Wagner, '96] [Mak-Wong, '00]

- **Min-*k*-Cut** in time $O(pn^{2k-1})$ for constant k
  [Chandrasekaran-Xu-Yu, '18]

Our main theorem

Hypergraph **Min-*k*-Cut** in time $O(mn^{2(k-1)})$ for constant k

- Implies **Min-Cut** in time $O(mn^2)$
- Improves upon [Chandrasekaran-Xu-Yu, '18] for **Min-*k*-Cut**

Random Contraction for Graph Cuts

RandomizedMinCut(G) [Karger, SODA '93]

Contract random edges until $n = 2$

Return the only cut

Random Contraction for Graph Cuts

RandomizedMinCut(G) [Karger, SODA '93]

Contract random edges until $n = 2$

Return the only cut

- Success probability: $\Omega(1/n^2)$. Run-time: $O(n^4)$
- For hypergraphs
  - uniform contraction doesn't work [Kogan-Krauthgamer, '15]
  - by non-uniform contraction
    - Min-Cut: $O(pn^2)$ [Ghaffari-Karger-Panigrahi, '18]
    - Min-k-Cut: $O(pn^{2k-1})$ [Chandrasekaran-Xu-Yu, '18]

Branching Contraction for Graph Cuts

RecursiveMinCut(G) [Karger-Stein, J. ACM '96]

G' ← Contract random edges until $n/\sqrt{2}$ vertices

Repeat twice

RecursiveMinCut(G')

Return the smaller cut from the 2 recursive calls

- Success probability: $\Omega(1/\log n)$. Run-time: $O(n^2)$.
- **Key challenge** for hypergraphs: contracting large edges
- Conceptual contribution of our work:

a Karger-Stein-type algorithm for hypergraphs

Non-branching Contraction for Hypergraph *k* Cuts

Let $z_n(e) = 1 - \frac{\binom{n-|e|}{k-1}}{\binom{n-1}{k-1}}$ be the rejection probability

Simple(*H*)

Repeat

Pick a random edge

With probability $1 - z_n(e)$, return **Simple**(*H/e*)

Non-branching Contraction for Hypergraph *k* Cuts

Let $z_n(e) = 1 - \frac{\binom{n-|e|}{k-1}}{\binom{n-1}{k-1}}$ be the rejection probability

> **Simple**(*H*)
>
> Repeat
>
> Pick a random edge
>
> With probability $1 - z_n(e)$, return **Simple**(*H/e*)

- An alternative interpretation of [Chandrasekaran-Xu-Yu, '18]
- To show: success probability is $\Omega(1/n^{2k-2})$ (for a fixed *k* cut)
  - Each run takes time O(*pn*)
  - Yields $O(pn^{2k-1})$ time bound [Chandrasekaran-Xu-Yu, '18]

Non-branching Contraction for Hypergraph k Cuts

Proof: Let q_n denote the minimum success probability over all n-vertex hypergraphs.

Show by induction

$$q_n \geq \frac{k}{\binom{n-1}{k-1}\binom{n}{k-1}}$$

— Success case 1: picks an edge, rejects, then succeeds

— Success case 2: picks an edge not in the cut, contracts, then succeeds

Non-branching Contraction for Hypergraph *k* Cuts

— **Success case 1**: picks an edge, rejects, then succeeds

Happens with probability $$\left(\frac{1}{m}\sum_e z_n(e)\right) q_n$$

— **Success case 2**: picks an edge **not in the cut**, contracts, then succeeds.

Lemma: Let *C* be a minimum *k*-cut. We have

[Chandrasekaran-Xu-Yu, '18]

$$|C| \leq \sum_e z_n(e)$$

Non-branching Contraction for Hypergraph *k* Cuts

— Success case 2: picks an edge not in the cut, contracts, then succeeds.

Lemma: Let C be a minimum k-cut. We have

[Chandrasekaran-Xu-Yu, '18]

$$|C| \leq \sum_{e} z_n(e)$$

Non-branching Contraction for Hypergraph *k* Cuts

— Success case 2: picks an edge not in the cut, contracts, then succeeds.

Lemma: Let C be a minimum k-cut. We have
[Chandrasekaran-Xu-Yu, '18]

$$|C| \leq \sum_{e} z_n(e)$$

Implies that a random edge is not in the cut w.p. at least

$$1 - \frac{1}{m} \sum_{e} z_n(e)$$

Non-branching Contraction for Hypergraph *k* Cuts

— Success case 2: picks an edge not in the cut, contracts, then succeeds.

Lemma: Let C be a minimum k-cut. We have
[Chandrasekaran-Xu-Yu, '18]

$$|C| \leq \sum_{e} z_n(e)$$

Implies that a random edge is not in the cut w.p. at least

$$1 - \frac{1}{m} \sum_{e} z_n(e)$$

Contraction occurs and the algorithm then succeeds w.p.

$$\frac{1}{m - |C|} \sum_{e} (1 - z_n(e)) \cdot q_{n-|e|+1}$$

Non-branching Contraction for Hypergraph k Cuts

Adding case 1 and case 2

$$q_n \geq \left(\frac{1}{m} \sum_e z_n(e) \right) q_n + \left(1 - \frac{1}{m} \sum_e z_n(e) \right) \frac{1}{m - |C|} \sum_e (1 - z_n(e)) \cdot q_{n-|e|+1}$$

Solving the recurrence yields

$$q_n \geq \frac{k}{\binom{n-1}{k-1}\binom{n}{k-1}}$$

Branching Contraction for Hypergraph k Cuts

Let $z_n(e) = 1 - \frac{\binom{n-|e|}{k-1}}{\binom{n-1}{k-1}}$ be the branching probability

Branching Contraction for Hypergraph *k* Cuts

Let $z_n(e) = 1 - \frac{\binom{n-|e|}{k-1}}{\binom{n-1}{k-1}}$ be the **branching** probability

Branching(*H*)
Pick a random edge
With probability $1 - z_n(e)$, return **Branching**(*H*/*e*)
Otherwise, **Branching**(*H*/*e*) and **Branching**(*H*)
Return the smaller cut from the *2* recursive calls

Branching Contraction for Hypergraph *k* Cuts

Let $z_n(e) = 1 - \frac{\binom{n-|e|}{k-1}}{\binom{n-1}{k-1}}$ be the **branching** probability

Branching(*H*)

Pick a random edge

With probability $1 - z_n(e)$, return **Branching**(*H/e*)

Otherwise, **Branching**(*H/e*) and **Branching**(*H*)

Return the smaller cut from the *2* recursive calls

Theorem

For constant *k*, the success probability is Ω(*1*/log *n*)

Branching Contraction for Hypergraph *k* Cuts

Let $z_n(e) = 1 - \frac{\binom{n-|e|}{k-1}}{\binom{n-1}{k-1}}$ be the **branching** probability

Branching(*H*)
Pick a random edge
With probability $1 - z_n(e)$, return **Branching**(*H*/*e*)
Otherwise, **Branching**(*H*/*e*) and **Branching**(*H*)
Return the smaller cut from the *2* recursive calls

Theorem
For constant *k*, the success probability is Ω(*1*/log *n*)

Proof: **Branching**(*H*/*e*) succeeds +
Branching(*H*) is called and succeeds — both succeed

Run-time Analysis: Random Computation Tree

- Each computation node is a hypergraph
- Creating a node of n_0 vertices takes time $O(mn_0)$

Let $S(n, n_0)$ be the number of computation nodes of n_0 vertices, starting with a hypergraph of n vertices

Lemma: For any n_0

$$S(n, n_0) \le \frac{\binom{n}{k-1}\binom{n-1}{k-1}}{\binom{n_0}{k-1}\binom{n_0-1}{k-1}} = O\left(\left(\frac{n}{n_0}\right)^{2k-2}\right)$$

By the lemma, runtime is bounded by

$$\sum_{n_0=k}^{n} O\left(\left(\frac{n}{n_0}\right)^{2k-2}\right) \cdot O(mn_0) = \widetilde{O}\left(mn^{2k-2}\right)$$

More Results

More Results

Low-rank hypergraphs (rank = max edge size)

Theorem
Min-k-Cut in time $O(n^{\max\{r, 2k-2\}})$ for hypergraphs of constant rank r.

Also follows a probabilistic branching contraction scheme

More Results

Low-rank hypergraphs (rank = max edge size)

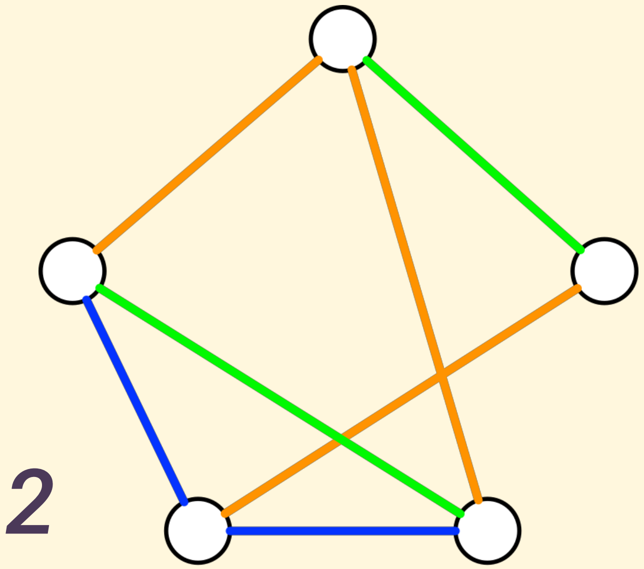
> **Theorem**
> **Min-*k*-Cut** in time $O(n^{\max\{r, 2k-2\}})$ for hypergraphs of constant rank r.

Also follows a probabilistic branching contraction scheme

Hedgegraphs [Ghaffari-Karger-Panigrahi, ’18]

(span = number of connected components induced by the hedge)

> **Theorem**
> **Min-Hedge-*k*-Cut** in time $O(m^2 n^{(s+1)(k-1)})$ for hedgegraphs of constant span s.



Improves upon [Chandrasekaran-Xu-Yu, ’18] for $k, s \geq 2$