Hypergraph $k$-Cut in Randomized Polynomial Time

Karthikeyan Chandrasekaran, Chao Xu and Xilin Yu
University of Illinois, Urbana-Champaign
Hypergraph and $k$-cut

Equivalently, set of edges whose removal disconnects the hypergraph into at least $k$ components.
Hypergraph and $k$-cut

$k$-cut: edges crossing a $k$-partition of vertices
Equivalently, set of edges whose removal disconnects the hypergraph into at least $k$ components
Hypergraph and $k$-cut

$k$-cut: edges crossing a $k$-partition of vertices

Equivalently, set of edges whose removal disconnects the hypergraph into at least $k$ components
Hypergraph and $k$-cut

$k$-cut: edges crossing a $k$-partition of vertices
Equivalently, set of edges whose removal disconnects the hypergraph into at least $k$ components
The hypergraph $k$-cut problem

- Given: Hypergraph $G = (V, E)$
- Output: Minimum cardinality $k$-cut
Applications of $k$-cut

- Network reliability
- VLSI design
- Clustering
- …
Previous works on GRAPH $k$-cut
Previous works on \textit{GRAPH} $k$-cut

- Reduction to min $st$-cut using uncrossing arguments: $n^{\Theta(k^2)}$
  
  [Goldschmidt-Hochbaum 94]
Previous works on **GRAPH** $k$-cut

- Reduction to min $st$-cut using uncrossing arguments: $n^{\Theta(k^2)}$  
  [Goldschmidt-Hochbaum 94]
- Randomized contraction: $\tilde{O}(n^{2(k-1)})$  [Karger-Stein 96]
Previous works on GRAPH $k$-cut

- Reduction to min $st$-cut using uncrossing arguments: $n^{\Theta(k^2)}$
  [Goldschmidt-Hochbaum 94]
- Randomized contraction: $\tilde{O}(n^{2(k-1)})$ [Karger-Stein 96]
- Divide and conquer: $O(n^{(4+o(1))k})$ [Kamidoi-Yoshida-Nagamochi 07]
- Divide and conquer: $O(n^{(4-o(1))k})$ [Xiao 08]
Previous works on **GRAPH k-cut**

- Reduction to min \(st\)-cut using uncrossing arguments: \(n^{\Theta(k^2)}\)  
  [Goldschmidt-Hochbaum 94]
- Randomized contraction: \(\tilde{O}(n^{2(k-1)})\)  
  [Karger-Stein 96]
- Divide and conquer: \(O(n^{(4+\omega(1))k})\)  
  [Kamidoi-Yoshida-Nagamochi 07]
- Divide and conquer: \(O(n^{(4-\omega(1))k})\)  
  [Xiao 08]
- Tree packing: \(\tilde{O}(n^{2k})\)  
  [Thorup 08]
Previous works on HYPERGRAPH $k$-cut

- Bipartite representation and max flow
- Vertex ordering
- Randomized contraction
- Deterministic contraction
- Constant rank: Hypertree packing

Hypergraph $k$-cut for $k \geq 4$ in arbitrary rank hypergraphs?
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
  - Vertex ordering [Klimmek-Wagner 96, Queyranne 98, Mak-Wong 00]
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
  - Vertex ordering [Klimmek-Wagner 96, Queyranne 98, Mak-Wong 00]
  - Randomized contraction [Ghaffari-Karger-Panigrahi 17]
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
  - Vertex ordering [Klimmek-Wagner 96, Queyranne 98, Mak-Wong 00]
  - Randomized contraction [Ghaffari-Karger-Panigrahi 17]

- $k = 3$:
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
  - Vertex ordering [Klimmek-Wagner 96, Queyranne 98, Mak-Wong 00]
  - Randomized contraction [Ghaffari-Karger-Panigrahi 17]
- $k = 3$:
  - Deterministic contraction [Xiao 08]
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
  - Vertex ordering [Klimmek-Wagner 96, Queyranne 98, Mak-Wong 00]
  - Randomized contraction [Ghaffari-Karger-Panigrahi 17]
- $k = 3$:
  - Deterministic contraction [Xiao 08]
- Constant rank: Hypertree packing [Fukunaga 10]
  (Rank of a hypergraph: size of the largest hyperedge)
Previous works on HYPERGRAPH $k$-cut

- $k = 2$, the hypergraph min-cut problem:
  - Bipartite representation and max flow
  - Vertex ordering [Klimmek-Wagner 96, Queyranne 98, Mak-Wong 00]
  - Randomized contraction [Ghaffari-Karger-Panigrahi 17]
- $k = 3$:
  - Deterministic contraction [Xiao 08]
- Constant rank: Hypertree packing [Fukunaga 10]
  (Rank of a hypergraph: size of the largest hyperedge)

Hypergraph $k$-cut for $k \geq 4$ in arbitrary rank hypergraphs?
Our result

Theorem

There exists a randomized polynomial time algorithm to solve the hypergraph k-cut problem.
$k = 2$: Hypergraph cut (arbitrary rank)
Contractions in hypergraphs
Contractions in hypergraphs
Edges in all cuts should not be contracted
Uniform probability contraction

Large probability of failure in a single step
Destroys the min-cut with one.pnum/two.pnum probability
one.pnum/two.pnum probability of success
Unclear how to analyze

\( n^3 \)
Uniform probability contraction

- Large probability of failure in a single step

\[ n^3 \]

\[ n^3 \]

\[ n^3 \]
Uniform probability contraction

- Large probability of failure in a single step
- Destroys the min-cut with 1/2 probability
Uniform probability contraction

- Large probability of failure in a single step
- Destroys the min-cut with 1/2 probability
- 1/2 probability of success
Uniform probability contraction

- Large probability of failure in a single step
- Destroys the min-cut with 1/2 probability
- 1/2 probability of success
- Unclear how to analyze
Our algorithm for hypergraph cut

Dampening factor:

\[ \delta_e := \Pr_{v \sim V}(v \notin e) = \frac{n - |e|}{n} \]
Our algorithm for hypergraph cut

Dampening factor:

\[
\delta_e := \Pr_{v \sim V} (v \notin e) = \frac{n - |e|}{n}
\]

**Input:** Hypergraph \(G\)

While there are more than 4 vertices in \(G\):

1. If \(\sum_{e \in E} \delta_e = 0\), return \(E\)
2. **Dampened sampling:** Pick \(e \in E\) with probability \(p_e := \frac{\delta_e}{\sum_{f \in E} \delta_f}\)
3. \(G \leftarrow G/e\)

Return a random min-cut in \(G\) by brute force
Analysis: Success probability

\[ q_n := \min_{C^* \in \text{OPT}(G)} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction
Analysis: Success probability

\[ q_n := \min \limits_{C^* \in \text{OPT}(G)} \text{Pr(Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[ q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1} \]
Analysis: Success probability

\[ q_n := \min_{C^* \in \text{OPT}(G)} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

\[ n \text{ node hypergraph } G \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[ q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1} \]

\[ = \sum_{e \in E \setminus C^*} \frac{\delta_e}{\sum_{f \in E} \delta_f} \cdot q_{n-|e|+1} \]
Analysis: Success probability

\[ q_n := \min_{C^* \in \text{OPT}(G)} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[ q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1} \]

\[ = \sum_{e \in E \setminus C^*} \frac{\delta_e}{\sum_{f \in E} \delta_f} \cdot q_{n-|e|+1} \]

\[ = \frac{1}{\sum_{f \in E} \delta_f} \sum_{e \in E \setminus C^*} \delta_e \cdot q_{n-|e|+1} \]
Analysis: Success probability

\[ q_n := \min_{C^* \in \text{OPT}(G)} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[
q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1} \\
= \sum_{e \in E \setminus C^*} \frac{\delta_e}{\sum_{f \in E} \delta_f} \cdot q_{n-|e|+1} \\
= \frac{1}{\sum_{f \in E} \delta_f} \sum_{e \in E \setminus C^*} \delta_e \cdot q_{n-|e|+1}
\]

For \( n > 4 \) and \( n - |e| + 1 \geq 2 \),

\[
\delta_e \cdot q_{n-|e|+1} \geq \left( \frac{n-|e|}{n} \right) \left( \frac{1}{\binom{n-|e|+1}{2}} \right)
\]
Analysis: Success probability

\[ q_n := \min_{C^* \in \text{OPT}(G)} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[
q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1}
\]

\[
= \sum_{e \in E \setminus C^*} \frac{\delta_e}{\sum_{f \in E} \delta_f} \cdot q_{n-|e|+1}
\]

\[
= \frac{1}{\sum_{f \in E} \delta_f} \sum_{e \in E \setminus C^*} \delta_e \cdot q_{n-|e|+1}
\]

For \( n > 4 \) and \( n - |e| + 1 \geq 2 \),

\[
\delta_e \cdot q_{n-|e|+1} \geq \left( \frac{n - |e|}{n} \right) \left( \frac{1}{\binom{n-|e|+1}{2}} \right)
\]

\[
= \frac{2}{n(n - |e| + 1)}
\]
Analysis: Success probability

\[ q_n := \min_{C^* \in \text{OPT}(G)} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[ q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1} \]

\[ = \sum_{e \in E \setminus C^*} \frac{\delta_e}{\sum_{f \in E} \delta_f} \cdot q_{n-|e|+1} \]

\[ = \frac{1}{\sum_{f \in E} \delta_f} \sum_{e \in E \setminus C^*} \delta_e \cdot q_{n-|e|+1} \]

For \( n > 4 \) and \( n - |e| + 1 \geq 2 \),

\[ \delta_e \cdot q_{n-|e|+1} \geq \left( \frac{n - |e|}{n} \right) \left( \frac{1}{\binom{n-|e|+1}{2}} \right) \]

\[ = \frac{2}{n(n - |e| + 1)} \]

\[ \geq \frac{1}{\binom{n}{2}} \]
Analysis: Success probability

\[ q_n := \min_{\begin{subarray}{c}
C^* \in \text{OPT}(G) \\
n \text{ node hypergraph G}
\end{subarray}} \Pr(\text{Algorithm returns } C^* \text{ on input } G) \]

Will show: \( q_n \geq \frac{1}{\binom{n}{2}} \) by induction

\[
q_n \geq \sum_{e \in E \setminus C^*} p_e \cdot q_{n-|e|+1}
\]

\[
= \sum_{e \in E \setminus C^*} \frac{\delta_e}{\sum_{f \in E} \delta_f} \cdot q_{n-|e|+1}
\]

\[
= \frac{1}{\sum_{f \in E} \delta_f} \sum_{e \in E \setminus C^*} \delta_e \cdot q_{n-|e|+1}
\]

\[
\geq \left( \frac{|E \setminus C^*|}{\sum_{f} \delta_f} \right) \left( \frac{1}{\binom{n}{2}} \right)
\]

For \( n > 4 \) and \( n - |e| + 1 \geq 2 \),

\[
\delta_e \cdot q_{n-|e|+1} \geq \left( \frac{n - |e|}{n} \right) \left( \frac{1}{\binom{n-|e|+1}{2}} \right)
\]

\[
= \frac{2}{n(n - |e| + 1)}
\]

\[
\geq \frac{1}{\binom{n}{2}}
\]
To show:

\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]
To show:

\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)
To show:
\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly
To show:
\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly

\(F_v\) be the edges containing \(v\), i.e., the \(v\) isolating cut
To show:
\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly

\(F_v\) be the edges containing \(v\), i.e., the \(v\) isolating cut

\(|C^*| \leq \mathbb{E}_{v \sim V} (|F_v|)\)
To show:
\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly

\(F_v\) be the edges containing \(v\), i.e., the \(v\) isolating cut

\[
|C^*| \leq \mathbb{E}_{v \sim V} (|F_v|) = \sum_{e \in E} \Pr_{v \sim V} (e \in F_v)
\]
To show:
\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \( |C^*| \leq |E| - \sum_{f \in E} \delta_f \)

Sample a vertex \( v \) from \( V \) uniformly

\( F_v \) be the edges containing \( v \), i.e., the \( v \) isolating cut

\[
|C^*| \leq \mathbb{E}_{v \sim V} (|F_v|) = \sum_{e \in E} \Pr(e \in F_v) = \sum_{e \in E} \left( 1 - \Pr(e \notin F_v) \right)
\]
To show:
\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly

\(F_v\) be the edges containing \(v\), i.e., the \(v\) isolating cut

\[
|C^*| \leq \mathbb{E}_{v \sim V} (|F_v|) = \sum_{e \in E} \Pr_e (e \in F_v) = \sum_{e \in E} \left(1 - \Pr_{v \sim V} (e \notin F_v)\right)
\]

\[
= \sum_{e \in E} \left(1 - \Pr_{v \sim V} (v \notin e)\right)
\]
To show:

\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly

\(F_v\) be the edges containing \(v\), i.e., the \(v\) isolating cut

\[
|C^*| \leq \mathbb{E}_{v \sim V} (|F_v|) = \sum_{e \in E} \Pr_{v \sim V} (e \in F_v) = \sum_{e \in E} \left(1 - \Pr_{v \sim V} (e \notin F_v)\right) = \sum_{e \in E} \left(1 - \Pr_{v \sim V} (v \notin e)\right) = \sum_{e \in E} (1 - \delta_e)
\]
To show:

\[
\frac{|E| - |C^*|}{\sum_{f \in E} \delta_f} \geq 1
\]

We show \(|C^*| \leq |E| - \sum_{f \in E} \delta_f\)

Sample a vertex \(v\) from \(V\) uniformly

\(F_v\) be the edges containing \(v\), i.e., the \(v\) isolating cut

\[
|C^*| \leq \mathbb{E}_{v \sim V} (|F_v|) = \sum_{e \in E} \Pr (e \in F_v) = \sum_{e \in E} \left(1 - \Pr (e \notin F_v)\right)
\]

\[
= \sum_{e \in E} \left(1 - \Pr (v \notin e)\right)
\]

\[
= \sum_{e \in E} (1 - \delta_e)
\]

\[
= |E| - \sum_{f \in E} \delta_f
\]
Theorem

The probability that the algorithm returns a particular min-cut is \( \frac{1}{\binom{n}{2}} \). Repeat it \( O(n^2 \log n) \) times to obtain a min-cut with high probability.
The algorithm for hypergraph $k$-cut

Similar algorithm as hypergraph cut, but different dampening.

$$\delta_e := \Pr_{s \sim \binom{V}{k-1}} (S \cap e = \emptyset) = \frac{\binom{n-|e|}{k-1}}{\binom{n}{k-1}}$$
The algorithm for hypergraph $k$-cut

Similar algorithm as hypergraph cut, but different dampening.

$$\delta_e := \Pr_{S \sim \binom{V}{k-1}} (S \cap e = \emptyset) = \frac{{n-|e| \choose k-1}}{{n \choose k-1}}$$

**Theorem**

The probability that the algorithm returns a particular min-$k$-cut is $\Omega\left(\frac{1}{n^{2(k-1)}}\right)$. 
The algorithm for hypergraph $k$-cut

Similar algorithm as hypergraph cut, but different dampening.

$$\delta_e := \Pr_{S \sim \binom{V}{k-1}} (S \cap e = \emptyset) = \frac{n-|e|}{\binom{n}{k-1}}$$

**Theorem**

The probability that the algorithm returns a particular min-$k$-cut is $\Omega\left(\frac{1}{n^2(k-1)}\right)$.

**Corollary of our algorithm and analysis**

The number of min-$k$-cuts is $O\left(n^{2(k-1)}\right)$. 
Additional Results: Hedgegraphs
Hedgegraphs

• A hedge is a collection of edges
• A hedgegraph consists of vertices \( V \) and a set of hedges on \( V \)
• The underlying graph of a hedgegraph is the union of its hedges
• Motivation: dependent edge failures
• Applications: layered networks, supply chain networks, ...
Hedgegraphs

- A hedge is a collection of edges
Hedgegraphs

- A hedge is a collection of edges
- A hedgegraph consists of vertices $V$ and a set of hedges on $V$
Hedgegraphs

- A **hedge** is a collection of edges
- A **hedgegraph** consists of vertices $V$ and a set of hedges on $V$
- The **underlying graph** of a hedgegraph is the union of its hedges
Hedgegraphs

- A hedge is a collection of edges
- A hedgegraph consists of vertices $V$ and a set of hedges on $V$
- The underlying graph of a hedgegraph is the union of its hedges
- Motivation: dependent edge failures
Hedgegraphs

- A **hedge** is a collection of edges
- A **hedgegraph** consists of vertices $V$ and a set of hedges on $V$
- The **underlying graph** of a hedgegraph is the union of its hedges
- Motivation: dependent edge failures
- Applications: layered networks, supply chain networks, . . .
Hedges
The span of a hedge is the number of components induced by a hedge.
Span of the blue hedge is 2
Additional results

- Poly-time algorithm for $k$-cut in constant span hedgegraphs (Hypergraphs are equivalent to hedgegraphs with span 1 [Ghaffari-Karger-Panigrahi 17])
- PTAS for $k$-cut in arbitrary span hedgegraphs
Additional results

- Poly-time algorithm for $k$-cut in constant span hedgegraphs (Hypergraphs are equivalent to hedgegraphs with span 1 \cite{Ghaffari-Karger-Panigrahi 17})
- PTAS for $k$-cut in arbitrary span hedgegraphs

Thank You!