Cut query algorithms with star contraction

Simon Apers
(CNRS, IRIF)

joint work with
Y. Efron, P. Gawrychowski, T. Lee, S. Mukhopadhyay, D. Nanongkai

FILOFOCS, Tel-Aviv University, June 2022

Talk outline

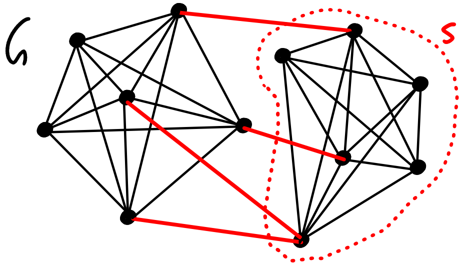
Talk outline

1. Cut queries and results
2. Star contraction
3. Quantum cut queries
4. Classical cut queries
5. Open questions

Cut query complexity

unweighted, undirected graph $G = (V, E)$

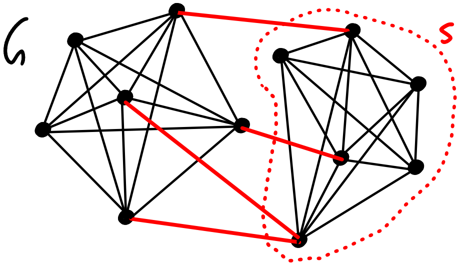
edge connectivity $\lambda(G) = \min_{\emptyset \neq S \subset V} |E(S, S^c)|$



Cut query complexity

unweighted, undirected graph $G = (V, E)$

edge connectivity $\lambda(G) = \min_{\emptyset \neq S \subset V} |E(S, S^c)|$

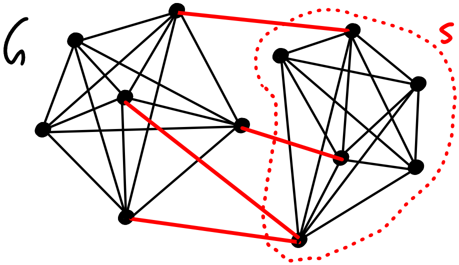


cut query $S \subseteq V \mapsto |E(S, S^c)|$

Cut query complexity

unweighted, undirected graph $G = (V, E)$

edge connectivity $\lambda(G) = \min_{\emptyset \neq S \subset V} |E(S, S^c)|$



cut query $S \subseteq V \mapsto |E(S, S^c)|$

? cut query complexity of $\lambda(G)$?

Cut query complexity

motivated

by submodular function minimization:

cut function $S \mapsto |E(S, S^c)|$ is submodular

Cut query complexity

motivated

by submodular function minimization:
cut function $S \mapsto |E(S, S^c)|$ is submodular

connected

to communication complexity,
streaming, matrix-vector queries, $\ldots$

Cut query complexity

some former results

Cut query complexity

some former results

- connectivity:

$O(n \log n)$ classical queries (Harvey '08)

$\widetilde{O}(1)$ quantum queries (Lee-Santha-Zhang, SODA'20)

Cut query complexity

some former results

- connectivity:

$O(n \log n)$ classical queries (Harvey '08)

$\widetilde{O}(1)$ quantum queries (Lee-Santha-Zhang, SODA'20)

- edge connectivity:

$O(n \log^3 n)$ classical queries (Rubinstein-Schramm-Weinberg, ITCS'18)

$\widetilde{O}(n)$ classical queries for *weighted* graphs (Mukhopadhyay-Nanongkai, STOC'20)

Cut query complexity

some former results

- connectivity:

$O(n \log n)$ classical queries (Harvey '08)
$\widetilde{O}(1)$ quantum queries (Lee-Santha-Zhang, SODA'20)

- edge connectivity:

$O(n \log^3 n)$ classical queries (Rubinstein-Schramm-Weinberg, ITCS'18)
$\widetilde{O}(n)$ classical queries for *weighted* graphs (Mukhopadhyay-Nanongkai, STOC'20)

our main results

Theorem

Cut query complexity

some former results

- connectivity:

$O(n \log n)$ classical queries (Harvey '08)

$\widetilde{O}(1)$ quantum queries (Lee-Santha-Zhang, SODA'20)

- edge connectivity:

$O(n \log^3 n)$ classical queries (Rubinstein-Schramm-Weinberg, ITCS'18)

$\widetilde{O}(n)$ classical queries for *weighted* graphs (Mukhopadhyay-Nanongkai, STOC'20)

our main results

Theorem

(i) The cut query complexity of (edge) connectivity is $O(n)$.

Cut query complexity

some former results

- connectivity:

$O(n \log n)$ classical queries (Harvey '08)

$\widetilde{O}(1)$ quantum queries (Lee-Santha-Zhang, SODA'20)

- edge connectivity:

$O(n \log^3 n)$ classical queries (Rubinstein-Schramm-Weinberg, ITCS'18)

$\widetilde{O}(n)$ classical queries for *weighted* graphs (Mukhopadhyay-Nanongkai, STOC'20)

our main results

Theorem

(i) The cut query complexity of (edge) connectivity is $O(n)$.
(ii) The quantum cut query complexity of edge connectivity is $\widetilde{O}(\sqrt{n})$.

Classical lower bound

k cut query algorithm implies $O(k \log n)$ communication complexity

Classical lower bound

k cut query algorithm implies $O(k \log n)$ communication complexity

hence,
$O(n)$ is optimal if communication complexity of connectivity is $\Omega(n \log n)$

Classical lower bound

k cut query algorithm implies $O(k \log n)$ communication complexity

hence,
$O(n)$ is optimal if communication complexity of connectivity is $\Omega(n \log n)$

which we conjecture,
but open since Babai-Frankl-Simon (FOCS'86)

Classical lower bound

k cut query algorithm implies $O(k \log n)$ communication complexity

hence,
$O(n)$ is optimal if communication complexity of connectivity is $\Omega(n \log n)$

which we conjecture,
but open since Babai-Frankl-Simon (FOCS'86)

best known (comm. compl.):
$\Omega(n)$ for connectivity (BFS'86)
$\Omega(n \log \log n)$ for edge connectivity (Assadi-Dudeja, SOSA'21)

Talk outline

Star contraction

Star contraction

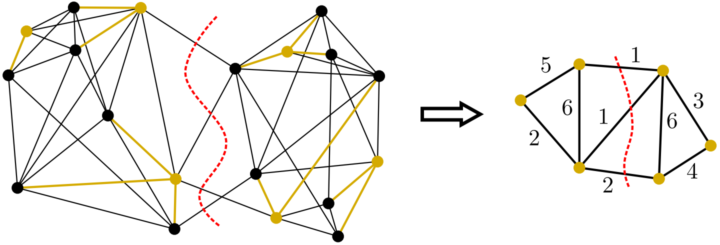
1. each vertex becomes center with probability $\widetilde{O}(1/\delta(G))$

Star contraction

1. each vertex becomes center with probability $\widetilde{O}(1/\delta(G))$
2. each remaining vertex merges with random neighboring center

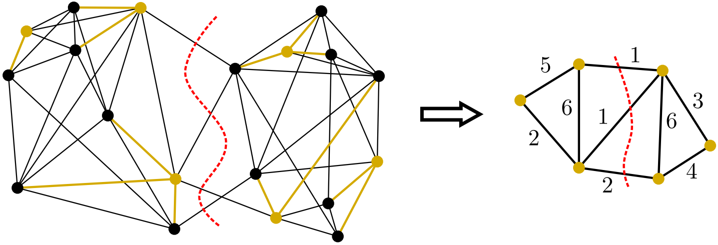
Star contraction

1. each vertex becomes center with probability $\widetilde{O}(1/\delta(G))$
2. each remaining vertex merges with random neighboring center



Star contraction

1. each vertex becomes center with probability $\widetilde{O}(1/\delta(G))$
2. each remaining vertex merges with random neighboring center

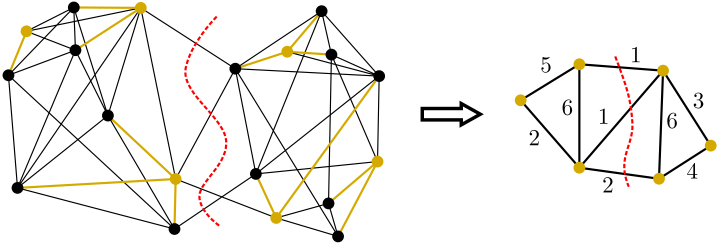


w.c.p., contracts to multigraph G' with

$|V(G')| \in \widetilde{O}(|V(G)|/\delta(G))$ and $\lambda(G') = \lambda(G)$ (if $\lambda(G) < \delta(G)$)

Star contraction

1. each vertex becomes center with probability $\widetilde{O}(1/\delta(G))$
2. each remaining vertex merges with random neighboring center



w.c.p., contracts to multigraph G' with

$$|V(G')| \in \widetilde{O}(|V(G)|/\delta(G)) \quad \text{and} \quad \lambda(G') = \lambda(G) \ \text{(if } \lambda(G) < \delta(G)\text{)}$$

related to “2-out contraction” (Ghaffari-Nowicki-Thorup, SODA’20)

Star contraction

Why does it preserve a minimum cut?

Star contraction

Why does it preserve a minimum cut?

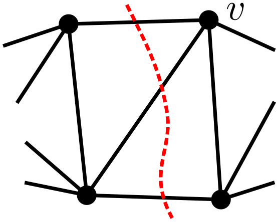
(assume vertices merge with random neighbor)

Star contraction

Why does it preserve a minimum cut?

(assume vertices merge with random neighbor)

vertex v can have $\leq d(v)/2$ edges across (nontrivial) min cut

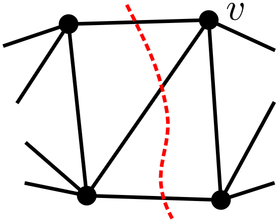


Star contraction

Why does it preserve a minimum cut?

(assume vertices merge with random neighbor)

vertex v can have $\leq d(v)/2$ edges across (nontrivial) min cut



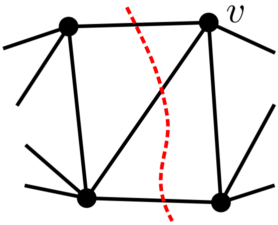
but at most $\delta(G) < d(v)$ edges in min cut

Star contraction

Why does it preserve a minimum cut?

(assume vertices merge with random neighbor)

vertex v can have $\leq d(v)/2$ edges across (nontrivial) min cut



but at most $\delta(G) < d(v)$ edges in min cut

so few vertices with large failure probability

Talk outline

1. Cut queries and results
2. Star contraction
3. Quantum cut queries
4. Classical cut queries
5. Open questions

Quantum cut queries

first studied by Lee-Santha-Zhang (SODA'21)
prove big separation with classical cut queries

Quantum cut queries

first studied by Lee-Santha-Zhang (SODA'21)
prove big separation with classical cut queries

rough idea: (generalized) cut query

$$|E(X,S)| = 1_X^T A_G 1_S = f_S(X)$$

evaluates linear function f_S

Quantum cut queries

first studied by Lee-Santha-Zhang (SODA'21)
prove big separation with classical cut queries

rough idea: (generalized) cut query

$$|E(X,S)| = 1_X^T A_G 1_S = f_S(X)$$

evaluates linear function f_S

$$\downarrow$$

using Bernstein-Vazirani,
can learn matrix-vector product $A_G 1_S$
using $\widetilde{O}(1)$ quantum cut queries

Quantum cut queries

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

Quantum cut queries

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

$\downarrow$

deal with high degree case through star contraction

Quantum cut queries

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

$\downarrow$

deal with high degree case through star contraction

2nd primitive (LSZ'21):
connectivity/spanning tree with $\widetilde{O}(1)$ quantum queries

Quantum cut queries

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

$\downarrow$

deal with high degree case through star contraction

2nd primitive (LSZ'21):
connectivity/spanning tree with $\widetilde{O}(1)$ quantum queries

$\downarrow$

deal with low degree case through "tree packing"

High degree case ($\delta(G) \geq \sqrt{n}$)

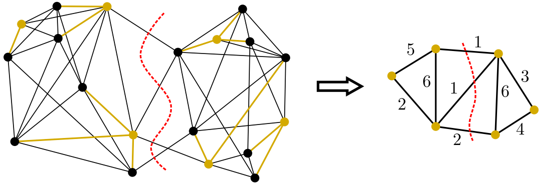
1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

High degree case ($\delta(G) \geq \sqrt{n}$)

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

$\downarrow$

star contraction:



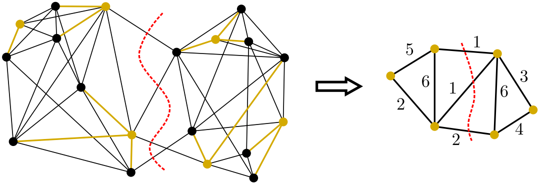
learn neighborhoods of $\widetilde{O}(\sqrt{n})$ centers with $\widetilde{O}(\sqrt{n})$ queries

High degree case ($\delta(G) \geq \sqrt{n}$)

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

$\downarrow$

star contraction:



learn neighborhoods of $\widetilde{O}(\sqrt{n})$ centers with $\widetilde{O}(\sqrt{n})$ queries

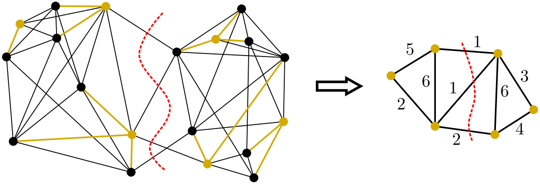
returns multigraph with $n/\delta(G) \in \widetilde{O}(\sqrt{n})$ vertices
can run *classical* cut query algorithm (MN'20)

High degree case ($\delta(G) \geq \sqrt{n}$)

1st primitive (LSZ'21):
learn all neighbors of a vertex with $\widetilde{O}(1)$ quantum queries

$\downarrow$

star contraction:



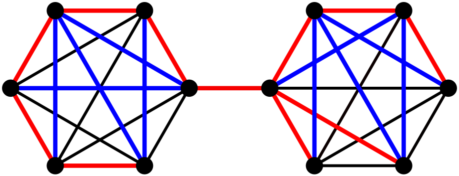
learn neighborhoods of $\widetilde{O}(\sqrt{n})$ centers with $\widetilde{O}(\sqrt{n})$ queries*

returns multigraph with $n/\delta(G) \in \widetilde{O}(\sqrt{n})$ vertices
can run *classical* cut query algorithm (MN'20)

* unclear for GNT's 2-out contraction

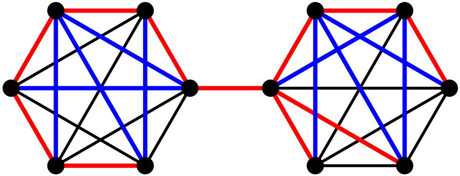
Low degree case ($\delta(G) < \sqrt{n}$)

"pack" spanning trees $F_1, F_2, \ldots$



Low degree case ($\delta(G) < \sqrt{n}$)

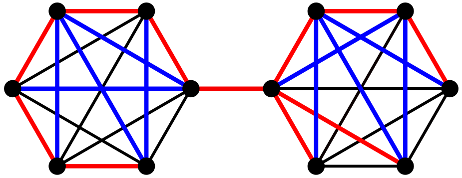
"pack" spanning trees $F_1, F_2, \ldots$



from max-flow min-cut theorem:
if $k \geq \lambda(G)$ then $\lambda(\cup_{i=1}^{k} F_i) = \lambda(G)$

Low degree case ($\delta(G) < \sqrt{n}$)

"pack" spanning trees $F_1, F_2, \ldots$



from max-flow min-cut theorem:
if $k \geq \lambda(G)$ then $\lambda(\cup_{i=1}^{k} F_i) = \lambda(G)$

algorithm:
pack $\sqrt{n}$ spanning trees = $\widetilde{O}(\sqrt{n})$ quantum cut queries
compute minimum cut of tree packing = no queries

Talk outline

Classical cut queries

gist:

Classical cut queries

gist:

(i) connectivity/spanning tree with $O(n)$ classical cut queries

Classical cut queries

gist:

(i) connectivity/spanning tree with $O(n)$ classical cut queries

(ii) star contraction with $O(n)$ queries

Classical cut queries

gist:

(i) connectivity/spanning tree with $O(n)$ classical cut queries

(ii) star contraction with $O(n)$ queries

(iii) returns $O(n/\delta(G))$ vertex graph, pack $\delta(G)$ trees with $O(n)$ queries

Connectivity

Harvey '08: connectivity with $O(n \log n)$ cut queries
based on Prim's (sequential) algorithm

Connectivity

Harvey '08: connectivity with $O(n \log n)$ cut queries
based on Prim's (sequential) algorithm
! uses only 1 out of $\log n$ bits of cut query

Connectivity

Harvey '08: connectivity with $O(n \log n)$ cut queries
based on Prim's (sequential) algorithm
! uses only 1 out of $\log n$ bits of cut query

$$\downarrow$$

squeeze out more using separating matrix: (Grebinski-Kucherov '98)
exists $O(n/\log n) \times n$ boolean matrix B such that

$$Bx \neq By, \quad \forall x \neq y \in \{0,1\}^n$$

Connectivity

Harvey '08: connectivity with $O(n \log n)$ cut queries
based on Prim's (sequential) algorithm
! uses only 1 out of $\log n$ bits of cut query

$$\downarrow$$

squeeze out more using separating matrix: (Grebinski-Kucherov '98)
exists $O(n/\log n) \times n$ boolean matrix B such that

$$Bx \neq By, \quad \forall x \neq y \in \{0,1\}^n$$

hence, can learn x from Bx (= $O(n/\log n)$ queries $y^T x$)

Connectivity

Harvey '08: connectivity with $O(n \log n)$ cut queries
based on Prim's (sequential) algorithm
! uses only 1 out of $\log n$ bits of cut query

$\downarrow$

squeeze out more using separating matrix: (Grebinski-Kucherov '98)
exists $O(n/\log n) \times n$ boolean matrix B such that

$$Bx \neq By, \quad \forall x \neq y \in \{0,1\}^n$$

hence, can learn x from Bx (= $O(n/\log n)$ queries $y^T x$)

$\downarrow$

implement Borůvka's (parallel) spanning tree algorithm
with $O(n/\log n)$ queries per round

$O(\log n)$ rounds so $O(n)$ total complexity

Talk outline

Open questions

- exponential quantum separations for *approximate* submodular function minimization?
  - e.g., *approximate* minimum cut with $\widetilde{O}(1)$ quantum cut queries?
  - ! can learn approximate min degree with $\widetilde{O}(1)$ quantum cut queries

Open questions

- exponential quantum separations for *approximate* submodular function minimization?
  - e.g., *approximate* minimum cut with $\widetilde{O}(1)$ quantum cut queries?
  - ! can learn approximate min degree with $\widetilde{O}(1)$ quantum cut queries
- Lower bounds:
  - $\Omega(n \log n)$ communication complexity of (edge) connectivity?
  - $\widetilde{\Omega}(\sqrt{n})$ quantum cut query complexity of min degree/cut?

Open questions

- exponential quantum separations for *approximate* submodular function minimization?
  - e.g., *approximate* minimum cut with $\widetilde{O}(1)$ quantum cut queries?
  - ! can learn approximate min degree with $\widetilde{O}(1)$ quantum cut queries
- Lower bounds:
  - $\Omega(n \log n)$ communication complexity of (edge) connectivity?
  - $\widetilde{\Omega}(\sqrt{n})$ quantum cut query complexity of min degree/cut?

Thank you!